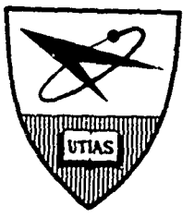
INSTITUTE
FOR
AEROSPACE STUDIES

UNIVERSITY OF TORONTO

AN ATTITUDE CONTROL SYSTEM TO CONSTRAIN THE SKIN TEMPERATURE OF A MANNED LIFTING SPACECRAFT DURING REENTRY INTO THE EARTH'S ATMOSPHERE

by

Jerome H. Fine



JULY 1967

UTIAS REPORT NO. 126
AFOSR 67-0858

AN ATTITUDE CONTROL SYSTEM TO CONSTRAIN THE SKIN TEMPERATURE OF A MANNED LIFTING SPACECRAFT DURING REENTRY INTO THE EARTH'S ATMOSPHERE

by

Jerome H. Fine

Manuscript received March 9, 1967.

JULY 1967

UTIAS REPORT NO. 126
AFOSR 67-0858

ACKNOWLEDGEMENTS

I express my sincere appreciation to Prof. B. Etkin for his help and guidance throughout the investigation and for the considerable time and effort which he expended on my behalf. I would also express my gratitude to Dr. G. N. Patterson for the opportunity to carry out the research at the UTIAS.

Special thanks are tendered to the staff at the University of Toronto Institute of Computer Science who operated the IBM 7094 and to Dr. C.C. Gottlieb who made the time available on the computer.

Financial support for this work was received from the United States Air Force, Office of Scientific Research, Applied Mathematics Division, under grants AF-AFOSR-222-63, 64, 66. The National Research Council of Canada provided support in the form of a scholarship.

SUMMARY

An attitude control system to regulate the temperature of a manned lifting spacecraft during reentry into the Earth's atmosphere is proposed. Its use prevents the peak skin temperature that is experienced during the reentry from rising moderately beyond that which would occur during an equilibrium glide of the same vehicle.

The effects of Earth rotation and oblateness upon the performance of the attitude control system were found to be moderate and predictable. The maximum temperature increment associated with them was found to be only 100 F^{o} for the worst set of initial conditions. The cross range shift of the footprint due to rotation was found to be within 70 miles of the value that would occur for the corresponding orbit in vacuum. Oblateness could generally be accounted for by using the effective initial glide angle relative to the Earth's surface rather than the geocentric initial value relative to the central coordinate system.

The results of density variations in the Earth's atmosphere were not serious. Large increases in the maximum skin temperature occurred only when extremely large spatially random density disturbances were encountered by the vehicle.

TABLE OF CONTENTS

NOMENCLATURE

Roman Letters

C_D	drag coefficient ($D/\frac{1}{2}\rho V^2 S$)
$C_{D_{min}}$	minimum drag coefficient (at zero angle of attack)
$C_{n,m}$	general spherical harmonic coefficient - see Eq. (4.1)
$C_{2,0}$	spherical harmonic coefficient associated with polar flattening
C_L	lift coefficient ($L/\frac{1}{2}\rho V^2 S$)
C_p	pressure coefficient in Newtonian flow - see Eq. (3.1)
D	aerodynamic drag on the vehicle (lb_f)
F	force (lb_f)
F_A	total aerodynamic force acting on the vehicle (lb_f)
f_c	defined by Eq. (4.17)
f_f	defined by Eq. (4.18)
f_p	the Earth's polar flattening $\left(\frac{R_o - R_{pole}}{R_o} = \frac{1}{298.28}\right)$
f_q	the Earth's equatorial flattening $\left(\frac{R_{major} - R_{minor}}{R_o} \approx 10^{-5}\right)$
f_r	defined by Eq. (4.16)
GM	the Earth's gravitational constant (ft^3/sec^2)
g	total acceleration due to the Earth's gravitational field (ft/sec^2)
g_r	acceleration in the direction of the geocentric radius - see Eq. (4.4a) (ft/sec^2)
g_λ	acceleration in the direction of the geocentric latitude - see Eq. (4.4b) (ft/sec^2)
g_θ	acceleration in the direction of the geocentric longitude - see Eq. (4.4c) (ft/sec^2)
$\{\bar{h},\bar{j},\bar{v}\}$	third intermediate reference system - see Fig. 21
I_α	vehicle moment of inertia about the pitch axis ($slug/ft^2$)
$\{\bar{I},\bar{J},\bar{K}\}$	basic reference system - see Fig. 18
$\{\bar{I}',\bar{J}',\bar{K}'\}$	first intermediate reference system - see Fig. 21
$\{\bar{i},\bar{j},\bar{k}\}$	flight axes reference system - see Fig. 20

K	parameter defined by Eq. (A.8)
k_α	angular acceleration in pitch (sec^{-2})
k_φ	angular acceleration in roll (sec^{-2})
L	aerodynamic lift on the vehicle (lb_f)
L/D	ratio of lift to drag
M	moment exerted by the pitch reaction jets (lb_f x ft)
m	mass of the vehicle (slugs)
$\frac{\overline{m}\,\overline{c}}{\overline{S}}$	heat capacity of the skin of the vehicle (BTU/ft^2)
$\{\overline{n},\overline{e},\overline{v}\}$	second intermediate reference system - see Fig. 21
P_n^m	general spherical harmonic - see Eq. (4.2)
Q	parameter defined by equation (3.9)
q	convective heating rate at any point on the vehicle (BTU/ft^2 sec)
q_n	nose stagnation point convective heating rate (BTU/ft^2 sec)
R_n	radius of curvature of the nose stagnation point (feet)
R_o	average equatorial radius (feet)
r	geocentric distance to the vehicle (feet)
S	reference wing area of the vehicle (ft^2)
$S_{n,m}$	general spherical harmonic coefficient - see Eq. (4.1)
s	distance along the flight path starting at the initial position (feet)
T	radiation-equilibrium nose stagnation point temperature (degrees Rankine or degrees Fahrenheit if specified)
$\hat{T}$	actual skin temperature at any point on the vehicle (degrees Rankine)
$T_{last_{max}}$	the higher of the present temperature and the values measured since testing for a local maximum started - see Fig. 12
$T_{last_{min}}$	the lower of the present temperature and the values measured since testing for a local minimum started - see Fig. 12
$T_{deadband}$	the allowable temperature variation before any action is taken - see Fig. 12

t	time starting at the initial point (seconds)
U	Earth's gravitational potential - see Eq. (4.1)
V	vehicle speed - see Fig. 19
y	perpendicular distance to the surface of the Earth from the vehicle - defined by Eq. (4.12) (feet)

Greek Letters

α	angle of attack of the vehicle (radians or if specified degrees)
γ	geocentric (with respect to the local geocentric horizontal) glide angle of the velocity vector - see Figs. 19 and 20 (radians or if specified degrees)
γ_e	initial geocentric glide angle - see Fig. 37
γ_{eff}	effective initial glide angle, defined by Eq. (B2) - see Fig. 37
Δt	defined by Eq. (2.4)
Δy	the difference in the perpendicular distance to the Earth's surface from the vehicle for the two latitude positions at the beginning and end of Phase II as measured at the average geocentric radius - see Fig. 37 (feet)
$\Delta\alpha$	defined by Eq. (2.6)
ϵ	emissivity of the material on the surface of the vehicle
θ	geocentric longitude of the position of the vehicle relative to the Prime Meridian - see Fig. 19 (radians or if specified degrees)
$\theta_{2,2}$	longitude of the equatorial major axis (radians or if specified degrees)
λ	geocentric latitude of the position of the vehicle - see Fig. 19 (radians or if specified degrees)
ρ	density of the atmosphere (slugs/ft^3)
σ	Stefan-Boltzmann constant (4.756×10^{-13} BTU/ft^2 sec $^\circ R^4$)
φ	angle of roll of the vehicle (radians or if specified degrees)
ψ	geocentric azimuth of the position of the vehicle - see Fig. 19 (radians or if specified degrees)
Ω	angular velocity of the $\{\bar{I},\bar{J},\bar{K}\}$ flight axes system relative to the $\{\bar{i},\bar{j},\bar{k}\}$ basic reference system (radians/sec)
ω	angular velocity of the Earth on its axis (radians/sec or if specified degrees/sec)

Superscripts

$\dot{(\ )}$	first derivative of a variable with respect to time
$\ddot{(\ )}$	second derivative of a variable with respect to time
$\bar{(\ )}$	denotes three-dimensional vector

Subscripts

$(\ )_{max}$	maximum value
$(\ )_{min}$	minimum value
$(\ )_s$	value when the last use of the reaction jets was started

I. INTRODUCTION

1.1 Historical Background

When the problems of manned space flight were being solved in the late 1950's and the early 1960's, the state of the art with regard to protecting reentry nose cones was oriented toward an ablative type of material since the heat pulse for ballistic missiles is characterized by intense heating rates which normally last for quite short intervals of time. Thus, the occurrence of manned space flight so soon after the successful implementation of the ballistic missile program produced technological solutions based primarily on concepts developed for the ICBM system. As a consequence, the heat shield concept which is presently associated with the manned capsule configuration seemed both a natural and logical extension of the techniques which were and still are generally accepted practice.

However, the capsule shape is able to attain only a very low ratio of lift to drag (L/D). Therefore, only very limited manoeuvring capability is available to the pilot. He is unable to land the spacecraft outside a relatively small area once the retro rockets have been fired and the deboost phase has been completed. As a result, only a small portion of the Earth's surface can be reached during any given twenty-four hour period. This is because each successive orbital track is displaced by about 22.5 degrees as a result of the Earth's rotation. Therefore, during that time period there would be only about sixteen passes over the whole Earth each yielding only a thin strip of coverage.

As a result of the above restrictions, a large pickup and recovery force must be maintained in the principal landing zone during the entire course of an exercise in case of an emergency. This procedure is unsatisfactory for many reasons. First, and perhaps the most obvious, is the sheer cost. However, and this reason is probably of greater importance, weather conditions may dictate the choice of landing area and a sudden storm might require the movement of the pickup force to a more suitable position within an impossibly short time interval. Other difficulties are also apparent upon even a preliminary analysis. The efficient solution to the landing problem appears to require a different shape of reentry vehicle. The basic requirement is greater manoeuvrability, which requires a higher L/D. Thus, the capsule configuration must be abandoned in favour of a winged vehicle or lifting body.

Another obvious requirement for space flight for the foreseeable future will be the minimization of space vehicle weight. It is not necessary to point out in any detail the well-known correlation between the cost of the first stage booster and the mass of the final payload. Therefore, the first step is to design a vehicle with a moderate L/D which possesses no excess protection to enable it to reenter the Earth's atmosphere and land on the Earth's surface. Considering the minimal degree of protection available, the basic flight path upon which the preliminary design is based should be a simple trajectory for which the problems of protecting the spacecraft and its occupants are as moderate as possible. However, when the choice for the basic path and the method of protection is being made, consideration must be given to incorporating provision for departures from the basic path. The departures are necessary to allow the vehicle to manoeuvre to landing sites other than the basic terminal position.

Before considering the choice of the vehicle and the basic path, two general areas should be reviewed. The concepts involved in protecting the

vehicle against extreme heating rates will be presented first. Then, the manoeuvring capability or 'landing footprint' available to a vehicle will be discussed.

1.2 Types of Protection Against Heating

There are two basic methods of withstanding reentry heating. Ablation materials have successfully been used on both ICBM nose cones and the current series of manned capsules in addition to other classes of vehicles. Very broadly speaking, the basic principle is to use a material which has a large heat of sublimation in order to absorb as large a total heat load as possible for a given mass. The actual heating rates experienced during the reentry are usually unimportant for such a design. However, the maximum heat load is very strictly defined. When exceeded, the heat shield will fail and the spacecraft will disintegrate.

On the other hand, a radiative material can withstand much higher total heat loads as long as the maximum heating rate does not exceed the capability of the material. The permissible heating rate is primarily a function of the temperature versus time capability of the material (see Fig. 1), its emissivity, and (for higher than average heating rates of a short duration) its heat capacity. If the material is thin enough, its heat capacity may be neglected and the temperature then assumes the radiation - equilibrium value. In this situation, the rate at which heat is absorbed locally by the skin of the spacecraft equals the rate at which heat is being radiated from its surface. Changes in the heating rate are then quickly reflected by a change in the skin temperature. A higher rate requires a higher temperature to radiate more heat and vice versa. In general, the position at which the heating rate is most severe is the nose stagnation point of the vehicle, since the nose cap has the smallest effective radius of curvature. Therefore, the nose stagnation point will have the highest temperature on the spacecraft. Thus, if a suitable material can be found for the nose stagnation point, it can safely be assumed that the rest of the vehicle can also be adequately protected. Consequently, unless explicitly stated, the radiation-equilibrium nose stagnation point temperature will be the value to which reference is being made whenever temperature is referred to in this paper.

Two other points which should be mentioned in comparing ablative and radiative cooling. The materials which employ the latter method must always be replaced after use. On the other hand, while recoating may be necessary for a severe reentry, it should be possible to reuse in a conventional manner a vehicle which employs radiation cooling. It should be noted, however, that ablative materials are able to handle very large heating rates (until the maximum heat load is reached) that are usually far larger than the heating rates which the maximum temperature of a radiation cooled material can support.

1.3 Footprint Terminology

In order to initiate a return to the Earth's surface, an orbiting spacecraft will fire its retro rockets. Once this deboost phase is over, the vehicle becomes a hypersonic glider which is capable of influencing the position of its landing site only by means of aerodynamic force controlled by the attitude of the vehicle. A representative manoeuvre boundary or footprint from Ref. 1 is shown in Fig. 2. Note that the initial position is at the altitude at which aerodynamic forces begin to affect the path of the reentry vehicle. Throughout this investigation, the computations were started when the perpendicular distance to the Earth's surface was 350,000 feet. As a consequence, the reentry

values of velocity and glide angle at the initial position were treated independently of the orbital altitude at which retro-fire takes place. The actual flight path is effectively separated for computational purposes from the point and conditions of retro-fire. The only restriction is that the deboost cannot occur at an altitude that is greater than the maximum height which is compatible with conditions at the initial position.

In Fig. 3, additional terminology is given regarding the projection of the footprint onto an imaginary sphere which is used in place of the Earth's surface. Geocentric spherical trigonometry is used to define the geographical position over the imaginary sphere. Since the footprint boundary is given for an altitude of about 100,000 feet, the usual conversion problems of positioning the vehicle in terms of its geodetic latitude and longitude and of determining the distance travelled over an oblate Earth can be avoided.

The actual method of presenting the results requires some explanation. Assume for the moment that the initial position of the vehicle is in the Earth's equatorial plane and that the initial azimuth is either due East or West. Note that the initial glide angle is quite immaterial to the range definition (although it is of great importance to the pilot since too large a value would cause the spacecraft to disintegrate through overheating). In this situation, the equatorial plane, the great circle projection, the orbital track projection, and the footprint midline all coincide. The 'down range' is the absolute value of the longitude change from the initial position (by definition, down range is positive). The cross range is the final latitude (cross range may be positive or negative) if the initial azimuth was due East and the negative of the final latitude if the initial azimuth was due West. Note that the definition must be consistant with the reentry trajectory track over the Earth's surface, not with the Earth's co-ordinate system. A third range parameter, the azimuth or heading angle change, may also be simply defined in this situation as the final azimuth minus (or plus in the case of a due West reentry) ninety degrees.

At the initial position (which has been arbitrarily defined to be the point over the Earth's surface when the vehicle's altitude is 350,000 feet), the initial values of speed, altitude, glide angle, azimuth, latitude, and longitude define the path which the vehicle will travel over the Earth's surface. For a non-rotating, spherical Earth, the final pattern or shape of the projection of the path on the Earth's surface will not be affected by the initial values of the last three parameters. The resultant equivalent down range, cross range and azimuth change must be the same as for the case of the due East equatorial reentry with the same initial values of speed, altitude, and glide angle. For this situation, the great circle projection, the orbital track projection, and the footprint midline will still coincide with each other, but they will no longer coincide with the equatorial plane. However, for a rotating, oblate Earth, all three will tend to diverge. In this situation, the great circle projection remains in the same relative position. Consequently, all range parameters are measured in equivalent degrees over the imaginary sphere relative to the great circle projection through the initial position. Figure 3 illustrates a typical footprint over a rotating, oblate Earth. The subroutine which computes these range values is listed in Appendix C under the name of SUBROUTINE RCNGE.

1.4 Choice of Vehicle and Basic Path

The type of vehicle chosen can be seen in Fig. 4. It satisfies the requirement of moderate lift to drag ratio since $(L/D)_{max}$ can range from 1.3 to 1.7 for this type of configuration.

With regard to the choice of the basic path, the manoeuvrability requirement is first used to determine the method of preventing the spacecraft from burning up during the reentry. When the vehicle must reach landing zones corresponding to large down range distances, prolonged flight times must result. Consequently the total heat input to an ablative material would be extremely large and might become prohibitive. However, in a design which radiates most of the heat away from the vehicle, the total heat load is not an important consideration. Therefore, a wide variation in the time interval over which the heating occurs will have a negligible effect on the vehicle design as long as the peak temperature experienced during the flight is independent of the point at which the spacecraft lands. Consequently, radiation cooling offers the only practical solution for protecting the vehicle against the extreme heating which is encountered during the reentry. This method then requires a trajectory that satisfies the requirement on the peak temperature.

An equilibrium glide was chosen as the basic path. This trajectory is characterized (see Appendix A for a more comprehensive discussion) by the absence of any long period or 'phugoid' oscillations along the trajectory. This situation results from the equality between the lift force and the difference between gravitational and centrifugal forces. There were a number of important reasons for the choice of the equilibrium glide. First, it satisfies the requirement that it be a simple trajectory since it is a fixed-attitude reentry. Second, it offers the lowest possible value for the peak temperature which is experienced by the vehicle during its reentry without the introduction of a sophisticated optimal programming procedure. Third, the equilibrium glide also satisfies the requirement that departures from this basic path may easily be made in order to provide the manoeuvring capability necessary to land at sites other than the basic terminal position.

The reader should also note the simple analytical result for the peak temperature which is experienced during an equilibrium glide (see Appendix A). The ease of obtaining this comparison standard and its simplicity of definition will greatly outweigh the benefits of obtaining any reduction in the standard peak temperature by complex optimal procedures.

1.5 CONTAC System

The problem, then, is to conceive of as simple a procedure as possible which will allow the next generation of reentry vehicles the freedom to manoeuvre to their desired landing zones. Ideally, implementation of this procedure will not require any additional protection for the pilot or the vehicle.

As a solution to this problem, the author proposes a CONstant Temperature Attitude Control (CONTAC) system which will limit the peak temperature which is experienced by the vehicle during a reentry to a value which is only moderately (by $20F^{o}$ or $30F^{o}$) greater than the peak temperature in a standard equilibrium glide. As a consequence, the CONTAC system will operate within the constraint that a negligible weight penalty is incurred relative to an

equilibrium glide, which it is assumed the vehicle can safely fly. A further constraint will be that neither predictive nor memory logic (with a minor exception) will be used by the CONTAC system. This requirement means that the CONTAC system must produce the solution on a real time basis when operating in the actual reentry environment.

It will be shown that the CONTAC system gives the pilot the extensive manoeuvrability which he requires in order to land the spacecraft at any point under or near the orbital envelope (Fig. 5). When coupled with the proper range controller, the complete system will provide automatic guidance to the final destination and prevent any dangerous temperature peaks which might destroy the vehicle. In addition, no undue waiting periods will be required. For most landing areas within the orbital envelope, the delay before retrofire will be less than twelve hours and no point will require more than twenty-four hours. For orbits of small inclination to the equatorial plane the waiting times would be usually less than one orbital period. Furthermore, the peak temperature will always be very near to the peak value encountered during an equilibrium glide except in those cases for which the atmospheric density variations about the mean are both large and random. Under these extreme conditions, it will be impossible to prevent a moderate rise in the peak temperature whenever the CONTAC system encounters a sequence of density variations for which no further correction by means of attitude control is available (i.e. when the optimum attitude - optimum angle of attack and zero roll angle - has already been assumed). Since adverse conditions which occur randomly are impossible to predict, a statistical result must be given and vehicle protection provided for the worst sequence of events which might occur with a certain design probability.

1.6 Comparison with Other Systems

As detailed in Ref. 1, a number of authors have studied the problem of guiding a spacecraft to its final destination. However, this investigation is not primarily concerned with the actual guidance, but rather with the means by which the heating problems may be minimized. In addition, the effects of Earth rotation and oblateness upon both the heating and the guidance problem will be ascertained. Finally, a test to determine the effect of random variations in atmospheric density will be presented.

With regard to the heating problem, a solution for minimizing the total heat load can be found in Refs. 2 and 3. However, since these solutions must operate within the inherent limitation imposed by the maximum allowable heat load, the manoeuvring capability is restricted to landing sites which are relatively close to the initial position.

The only other solution for minimizing peak temperatures that has been found in the open literature is a Temperature Rate Flight Control System which has been developed by Stalony-Dobrzanski (Ref. 4). The results show that the TRFCS is also able to prevent the peak temperature which occurs during a reentry from becoming greater than the value which is experienced during an equilibrium glide. (The use of the equilibrium glide as the basic path for the TRFCS is an additional reason for its choice as the standard of comparison in the present investigation.) However, in order to implement the TRFCS, the derivative of the temperature is required along with an elaborate analogue system to analyze the rate of temperature change. The attitude of the vehicle is adjusted using either the angle of attack or the bank angle as the primary control variable.

In contrast, the CONTAC system uses a simple digital program to analyze directly the actual temperature. In addition, the attitude is controlled via pitch and roll channels simultaneously.

With regard to the effects on the Earth's rotation and oblateness, there are no papers available which analyze the resulting changes in the footprint when constraints upon the peak temperature (or for that matter maximum heat load) are present. There have been a few studies which analyzed the effects of the Earth's rotation, but no constraints in connection with heating problems were incorporated.

Due to the lack of significant statistical data, the influence of density variations in the Earth's atmosphere is still extremely difficult to analyse. A number of average curves of density versus altitude are available for both summer and winter for latitudes from the equator to 75°N. Based on this type of information, a number of papers have appeared which deal with the effect of a constant percentage change in the density upon the guidance problem. However, even if heating effects were included in these investigations, the results cannot take account of the fact that the atmosphere is a dynamic system whose state can only be described in a statistical manner. With regard to the effects of random density variations about an average curve of density versus altitude, the author is not aware of any other studies which deal with the heating problem either from an optimal point of view (such as those represented by Refs. 2 and 3) or from a system approach (such as proposed by Stalony-Dobrzanski). Indeed, from a detailed examination of the way in which optimal procedures are applied, the inclusion of random density variations would seem to rule out this concept. The iteration procedure used depends on accurate external environmental specifications. If the density is a random quantity, it would not be possible under the presently used procedures to arrive at a final solution. Only a real time approach inherent in both the TRFCS and the CONTAC system can attempt to provide a solution. The present investigation will include the results of the attempts by the CONTAC system to deal with random density variations as well as constant percentage changes in the density.

1.7 Trajectory Phases

In order to simplify the description and analysis of the various aspects of the CONTAC system, the reentry trajectory has been divided into five rather distinct phases. Each phase is described rather briefly at this point in order to acquaint the reader with the overall concept. What the CONTAC system is capable of doing is also explained. In addition, problems which it cannot solve are mentioned and some suggestions are made regarding the solution of these difficulties.

It is assumed that the vehicle is under the primary control of a navigation and guidance system, that monitors and adjusts the two primary attitude angles, α and φ , in order to home on the desired landing site. At certain short but critical periods of the flight, the CONTAC system, as a result of monitoring vehicle temperature, automatically cuts in and overrides the navigation system. This situation occurs only during the portions of Phases III and IV of the reentry when the CONTAC system is in active operation. There are, of course, additional constraints on the attitude angles. These will be presented during the discussion of each phase.

The five phases of the trajectory are shown in Fig. 6. Although no such sharply defined divisions will occur during an actual flight, it is simpler to divide the reentry into the five parts shown for the purpose of explaining what takes place within each phase.

The CONTAC system has been designed for space vehicles entering the Earth's atmosphere from altitudes up to five hundred miles. It is conceivable that some application might be found for reentries from positions higher than this altitude. However, the primary requirement is that the maximum convective heating rate first experienced during the reentry (Fig. 6) must not produce a temperature greater than the peak value which would be experienced during an equilibrium glide. If this constraint is satisfied, the CONTAC system is capable of preventing all subsequent maximum temperatures from becoming very much greater than the peak experienced during an equilibrium glide. But, if the deboost altitude is above five hundred miles, the spacecraft must reenter the atmosphere at a speed which is substantially greater than the circular value. In order to prevent skipout, the reentry glide angle must also be increased. As a result, the first maximum experienced for the convective heating rate exceeds the peak for an equilibrium glide. Unless this single pulse can be tolerated, the vehicle fails.

Therefore, the CONTAC system is intended to be used for reentries from low orbital altitudes, up to about five hundred miles, with vehicles that use radiative cooling as the primary protection against failure due to heating. Examples of this type of operation include shuttle service for space stations, low orbit reconnaissance, and low orbit satellite service and repair.

1.8 Phase I - Pre-Atmospheric Flight

It should be noted at this point that although the CONTAC system may go into operation as soon as the retro rockets have been fired (or some other method is used to push the spacecraft out of its vacuum orbit), the angle of attack of the vehicle is quite immaterial for a period of between ten and thirty minutes, depending on the height of orbital ejection. This delay is due to the fact that aerodynamic forces do not become appreciable until an altitude of about 350,000 feet is reached (Fig.7). Thus the pilot will have ample time to jettison any service modules or extraneous externals and to perform the tasks associated with the alignment of the spacecraft prior to the buildup of atmospheric density.

One point of importance which is associated with this phase is the deboost attitude alignment and velocity increment. Both values must be carefully computed and equally carefully implemented since a large error in either could cause excessive heating rates leading to the failure of the spacecraft. Figure 8 shows the effect of errors in retro-fire attitude alignment and velocity increment. However, past experience in the Gemini program indicates that the require accuracies for attitude alignment and velocity increment will not be difficult to achieve. Interpretation of the results indicates that the Gemini retro-fire system experienced maximum errors of one degree in attitude alignment and three feet per second in velocity increment (Ref. 5). A simultaneous maximum error in both would yield a maximum resultant error at the end of Phase I at 350,000 feet of 0.1 degrees in flight path angle, 10 feet per second in speed and 2 degrees in down range. Cross range might also be affected but to a negligible extent; the maximum error could amount to only 0.2 degrees. Note that all of the above nominal error values are maximum and would be very unlikely to occur simultaneously.

Since experience has proven that a highly accurate calculation of the alignment attitude and velocity increment values is possible, there should be no difficulty in this area. However, it may be necessary to change the method by which the calculation is made. Consequently, the allowable margin of error is described and discussed in Section VI where the effects of incorrect reentry conditions at the beginning of Phase II are considered.

1.9 Phase II - Overall Down Range Control

For the purposes of this investigation, the reentry begins at 350,000 feet. This is due to the fact that for all practical purposes, the atmosphere has no effect on the flight path until the spacecraft reaches this altitude. Since the pilot has no control over the trajectory between the time he fires the retro rockets and the time he descends to this height, it is best to separate the problems associated with achieving the correct reentry conditions at 350,000 feet (by the proper attitude alignment and retro velocity increment at orbital altitudes) from the problems associated with controlling the vehicle's down range (by the proper angle of attack once lift and drag become appreciable). In fact, in order to simplify the problem even further, the portion of the flight within the atmosphere has been sub-divided into four parts, Phases II, III, IV, and V of the reentry.

During Phase II, the CONTAC system must be in operation; however, it exercises no actual control except to signal the end of Phase II at the point when the first q_{max} is reached (Fig. 9). At this position, the CONTAC system switches from passive to active operation and Phase III begins. Therefore, during Phase II, the pilot (or the automatic guidance system) has the function of devising the angle of attack program that will eliminate down range errors which may initially be present or which may occur during Phase II due to the Earth's departure from a circular, non-rotating sphere with a standard time invariant atmosphere. Since Phase II lasts for about four minutes, there is ample time for any down range corrections which are required in order to ensure that the vehicle lands on target.

To some extent, it will be possible to compensate, by adjustment of the initial retro-fire, for down range errors which it is anticipated will occur during Phase II due to some of the above non-ideal conditions. As a first approximation, a constant angle of attack can be used during the calculation for Phase II. In this situation, down range errors which will occur due to rotation and oblateness can be evaluated rather easily for this portion of the flight. The same calculation procedure can also be used to find the down range errors which will occur during Phase I. Thus, at the deboost position, it should be feasible to make the necessary corrections for down range errors which will occur during Phases I and II due to Earth rotation and oblateness. Note that the time of retro-fire may also be varied in addition to the attitude alignment and velocity increment. This additional parameter will probably provide most of the necessary correction.

Errors in down range which occur during Phase II due to variations in the atmospheric density are much more difficult to evaluate. Since it is unlikely that an accurate prediction of the density can be made, an estimate based on the available statistical data should be made. It may then be possible to compensate for the anticipated down range error by adjusting the retro-fire parameters.

The only restriction on the angle of attack during Phase II (angle of roll is zero) is that it must not be so low as to cause the first q_{max} to exceed the maximum allowable value nor so high as to endanger the rear portion of the vehicle from excessive heating rates. In general, the optimum value lies very near the angle of attack which yields the maximum lift coefficient. Larger values for the angle of attack will reduce the down range and smaller values will increase it. For the vehicle studied, the optimum angle of attack was 55 degrees. The largest angle of attack used during Phase II was 65 degrees and the smallest value was 35 degrees. These two limits yielded a down range interval of about 75 degrees or about 5000 nautical miles.

It should be noted that almost all of the incremental down range achieved by modulating the angle of attack during Phase II is not apparent until the end of Phase III. This is because the primary effect of varying the entry angle of attack is a variation in the vehicle's speed at the end of Phase II without any major difference in altitude, glide angle, or down range distance (Fig. 9). A comprehensive discussion and analysis of this point is given in Sec. 6.2.

1.10 Phase III - Equilibrium Glide Attainment

The most critical portion of the reentry is Phase III, the various sections of which are shown in Fig.10. During this part of the flight, the CONTAC system must remove energy from the vehicle at a rate designed to place the vehicle on or very near to an equilibrium glide trajectory. However, the rate of energy dissipation must also be flexible enough to allow the pilot (or the automatic guidance system) to control the vehicle's range in order to counteract all departures from non-ideal conditions as well as any errors present at the start of Phase III. Failure of the CONTAC-guidance combination to cope with adverse conditions will result in the vehicle overheating during the flight or landing at the wrong location, either of which could be disastrous.

The actual details of the CONTAC system are given in Section II. At this point, it is sufficient to state that the phugoid or long period oscillation can be completely eliminated before the peak temperature is experienced and as a result, that value is very nearly minimized.

For most reentries, the attitude which is adopted subsequent to the first q_{max} is held for between two and four minutes. During this period, the pilot again has an opportunity to continue correcting any down range errors as well as beginning to reduce any cross range discrepancy. The additional incremental down range available during Phase III varies from about 1500 miles when the entry angle of attack is 65 degrees up to about 5000 miles when the entry angle of attack is 35 degrees. It should be repeated that practically all of the down range increment available due to Phase II attitude control actually occurs during Phase III due to the variation in speed at the beginning of Phase III. The pilot or the guidance system must be aware of and be able to compensate for this speed difference.

During Phase III, a number of pairs of values are used for the upper and lower limits on the attitude angles of the vehicle. These pairs of values are inputs to the CONTAC system and may be regarded as constants within this system. During an actual or simulated reentry in which a pilot or range controller is available, the attitude limits will be changed to provide the

necessary range control in order to land the spacecraft at the proper point on the Earth's surface. It is a feature of the CONTAC system that it is unaffected by the values for these limits as long as the upper and lower values of certain key pairs do not coincide. As long as this requirement is satisfied, the CONTAC system can damp the long period oscillations from the trajectory and place the vehicle on the equilibrium glide before the peak temperature occurs.

In very simple terms, the pilot will select the upper and lower limits for the angles of attack and roll which best fit the choice of the landing area toward which he desires to proceed, although certain practical considerations will restrict the pilot's freedom of choice (the upper and lower limits must not coincide for both pitch and roll). The CONTAC system then automatically adjusts the attitude angles (or the pilot himself if the associated control hardware should fail, but not the temperature readout equipment) to some value between the upper and lower limits of the angles of pitch and roll.

It has definitely been determined that the CONTAC system will hold the temperature at very nearly a constant value as long as the vehicle's attitude has not reached either an upper or lower limit. A single skip usually occurs just after the first q_{max}, but this is quite intentional, and is designed to provide down range control capability during the early part of the reentry. Indeed, in some situations, this short skip is completely eliminated in order to substantially reduce the down range.

Thus, by the time the vehicle descends to about 260,000 feet, it has been placed upon an equilibrium glide corresponding to the attitude at which the spacecraft is currently being held. The lowest value for the peak temperature is achieved by using the optimum attitude. It occurs at maximum vertical force (see Appendix A) which is at 55 degrees angle of attack and zero roll angle. This optimum attitude is then held by the CONTAC system quite automatically until the peak temperature is attained. At that point, Phase IV begins and the attitude is changed to provide down range and cross range manoeuvring capability. In practically all the re-entries studied, Phase IV began at an altitude of about 245,000 feet. However, in cases where the density variations were quite large, Phase IV began at an altitude as much as 30,000 feet lower than the above value.

1.11 Phase IV - Attitude Control Release

There will, of course, be problems associated with unpredictable increases in air density which may drive the temperature during this phase beyond its previous maximum. In such a situation, it might be expected that the CONTAC system would have to revert to Phase III operation. However, the reader may recall that no such sharply defined division occurs during the actual flight. Therefore, since the CONTAC system has neither memory nor predictive logic, it is impossible for any such division to exist in the controlling concept. When in operation, the CONTAC system is attempting to keep the temperature constant, and if an increase occurs, whatever the cause, the CONTAC system will change the attitude so as to reduce the temperature. If the attitude is already at its optimum value and the CONTAC system is unable to prevent a further increase in the temperature, then the conclusion to be drawn is quite obvious, i.e. Phase III had not yet been completed. But the reversion is only with respect to the author's arbitrary division of the flight. The CONTAC system is not affected.

In fact, the pilot (or the automatic guidance system) need not in principle even know that the CONTAC system exists, let alone how it operates. Naturally, he would wonder why he is unable to assume a particular attitude for range control during certain portions of the reentry. However, the CONTAC system is completely automatic so that it can cope with unpredictable situations about which the pilot may be unaware.

The pilot would, of course, actually be informed about both the existence and the operation of the CONTAC system and he may, therefore, depend on it to properly reduce the attitude from the optimum position at the beginning of Phase IV to the desired values of pitch and roll sometime before the end of Phase IV. In addition, with regard to where Phase III ends and Phase IV begins, the pilot can leave this quite arbitrary decision entirely to the built-in logic of the CONTAC system. It is quite immaterial, as far as he is concerned, just where the peak temperature occurs during the flight. The only problem is to prevent that value from becoming any larger than necessary, and this problem the CONTAC system will solve all by itself.

Therefore, during Phase IV, the CONTAC system will gradually reduce the angles of pitch and roll until at some particular point in the flight, both values are under the complete control of the pilot. The transfer of this control may be revoked at any time after it is first made; however, after the temperature has fallen several hundred degrees, the pilot will realize that the transfer is complete and that the CONTAC system is no longer in operation. This transfer always occurs by the time the speed and height have dropped to values in the vicinity of 12,000 feet per second and 150,000 feet, respectively. At that time, a procedure to increase the cross range will have come into effect. A paper by Wagner (Ref. 6) discusses the variation of the bank angle of the spacecraft in relation to cross range control. The results indicate that large angles of roll should be used at the beginning of the reentry when the speed is high and that small angles of roll are best near the end of the flight when the speed is low. Various modes of dependence between the angle of roll and the flight parameters speed, altitude, and energy, were investigated to determine the best relationship. The calculations showed that a simple linear variation of bank angle versus speed gives the best overall performance.

1.12 Phase V - Final Range Control and Touchdown

As this paper is being written, the procedures for the final landing of a spacecraft are being developed in principle and tested in practice. However, an analysis of this phase is outside the scope of this investigation. It is therefore, of little value to speculate upon the best method to reach the landing site and perform the touchdown. Several papers (among them Refs. 7, 8, and 9) have appeared which deal with this manoeuvre so that some ideas are already available concerning the problems which may be encountered and the methods of solution.

To this date, all of the flight experience in the supersonic regime has been with the X-15 vehicle. Thus, it may be quite some time before the best solutions to the various problems are found. However a basic outline of the final phase has emerged from this test program. The additional local down range which can be attained is about 200 miles for the conditions given in Fig. 3 for the footprint boundary. But, it may be noted that the final azimuth is approximately perpendicular to the boundary line. Consequently, the footprints which are presented in Sections VI and VII are a conservative estimate of the

vehicle's capability. One point of considerable importance should be noted. At the end of Phase IV, the valuesof speed and altitude within and on the footprint boundary are independent of the final position. Consequently, the same procedure can always be used for the approach during Phase V.

II. ATTITUDE CONTROL LOGIC

2.1 Introduction

This section discusses the detailed concepts behind the CONTAC system. The methods by which the system may be implemented on the spacecraft itself will also be suggested although the actual details may require modifications in order to conform to hardware requirements.

In order to give the reader some insight as to why the CONTAC system is used, a typical reentry is shown in Fig. 11a in which no attitude control was attempted except at the first skip point. At this position, the angle of attack was reduced from the optimum value of 55 degrees and zero roll angle to 22.7 degrees with a roll angle of 50 degrees to the left in order to obtain greater down range and to produce some cross range. However, due to the reduction in the angle of attack, a phugoid oscillation is initiated which results in a large temperature oscillation as well. Because the latter is synchronized almost exactly with the phugoid, it was logical to attempt to control both the phugoid and the temperature oscillations by monitoring only the temperature. The reason for the use of the temperature as the controlling variable is that this value can be measured directly on board the spacecraft with a high degree of accuracy. On the other hand, the altitude is a very difficult quantity to measure when the vehicle is still 250,000 feet above the Earth's surface. In addition, the temperature oscillations are the ones which must be controlled. It is quite immaterial whether or not the phugoid oscillation is controlled and damped out as soon as possible or is amplified and made larger so long as the temperature peaks are reduced. Of course, the fact that the elimination of the temperature oscillation also results in the elimination of the phugoid (Fig. 11b) is an added bonus which results in greater range control over the vehicle by the pilot.

It may be of interest to note that attitude control based upon altitude measurements was also investigated at an early stage in the development of the CONTAC system. In view of the results, which were inferior to the CONTAC concept, and the difficulty associated with the measurement of the altitude during an actual reentry, all effort on this aspect was abandoned.

A discussion of the basic philosophy behind the CONTAC system will provide a greater understanding of how it works. The main purpose is to dissipate as much of the vehicle's kinetic energy at as high an altitude as possible. In order to achieve this result, the CONTAC system is designed to prevent any decrease in the temperature after the first maximum is reached during the initial pass into the atmosphere. In order to prevent the decrease, the angle of attack is reduced which increases the L/D (actually the ratio of vertical force to drag, but at constant bank angle, L/D is just as appropriate) and thereby extends the glide path over a longer distance at the higher altitudes. However, the angle of attack eventually reaches its minimum value and at that point the temperature will begin to fall. However, when at some later time the temperature again starts to rise, a decision must be made as to when is the best time to increase

the angle of attack in order to prevent a subsequent temperature peak from exceeding the highest temperature experienced prior to that point in the flight. With no prediction scheme available, the best choice of the time to increase the angle of attack is immediately, i.e. as soon as the temperature again begins to rise. Keeping the temperature constant from that point on implies a gradual reduction in L/D which in turn may result in lower altitudes sooner than is desirable. There may be an optimum period of time to wait before initiating any increase in the angle of attack. However the absence of both memory and predictive logic within the CONTAC system precludes any use of a time delay interval.

In summary, the CONTAC system operates on the principle that the best place to lose speed is at as high an altitude as possible. Therefore, the angle of attack is gradually reduced in such a manner as to prevent the temperature from falling after it achieves a local peak value. That action results in an extension of the glide path at higher altitudes due to the resulting higher values of L/D. On the other hand, it was found best (by increasing the angle of attack) to prevent for as long as possible any subsequent rise in temperature after a minimum value occurs. The result is a system which always attempts to keep the temperature constant as long as angle of attack changes are possible, i.e. as long as the angle of attack was at neither its lower nor upper limit following a maximum or minimum, respectively, in the equilibrium nose stagnation point temperature.

The same principles are also used to control the bank angle in conjunction with the angle of attack. The only difference is that the "wings level" position is substituted for the upper limit since the maximum value for the ratio of vertical force to drag occurs for zero bank angle. On the other hand, the minimum value for this ratio occurs at the largest angle of roll. Therefore, the "lower limit" is the largest bank angle to the right or left depending on which direction the landing zone is situated with respect to the orbital track.

2.2 Choice of Attitude Limits

There are certain practical limitations on the freedom of choice for the upper and lower limits for both the angle of attack and the bank angle. Most of the limitations are due to the problems associated with obtaining as small a value as possible for the peak temperature experienced during the reentry of the spacecraft. In addition, the allowable range of values for the upper and lower limits changes with each phase of the reentry. Consequently, the best procedure will be to take each phase in turn and describe the reasons for the restrictions during that part of the trajectory.

(a) During Phase I, there are no restrictions and no practical limitations since the atmosphere is not effective in controlling the trajectory. The only point at which the attitude must be controlled is at the beginning of Phase 1 during the retro-fire. At that time, the attitude control must be quite precise, but there is, of course, no restriction on the choice of the angles. Near the end of Phase I, however, the angle of attack should be somewhere near the value to be used during Phase II since, otherwise, stability and control problems might be encountered.

(b) During Phase II, the bank angle should always be zero in order to produce the maximum possible vertical force for a given angle of attack. It might be feasible to introduce some aerodynamic side force on the vehicle at this

time, but the results would hardly be worth the added complexity necessary in the guidance system. This is because only large angles of roll would really be effective in producing cross range at this high a speed (Ref. 6), but the associated reduction in vertical force could not be tolerated.

The angle of attack should always be near the optimum value of 55 degrees which gives the maximum lift coefficient. In the presence of a predetermined atmosphere in which there were no density variations about the mean curve of density versus altitude, it would be possible to determine the desired angle of attack which would then be held throughout Phase II. However, in general, the guidance system will probably be making small changes throughout Phase II in order to compensate for predicted errors in the terminal down range position. Since the CONTAC system does not go into active operation until the end of Phase II, the angle of attack is completely under control of the guidance system during this time.

The largest angle of attack which was used herein during PhaseII was 65 degrees. However, values as high as 75 degrees would probably be feasible if the spacecraft could withstand the resulting higher heating rates at the rear of the undersurface of the vehicle. The smallest value used was 35 degrees. When still smaller values were tried, the first maximum temperature was higher than the peak temperature for an equilibrium glide.

The designation of the upper limit on the attitude (no lower limit was used) was the entry angle of attack (which varied between 35 degrees and 65 degrees) and the entry angle of roll (which was always zero).

(c) During Phase III, the CONTAC system is in active operation about half the time. The vehicle is initially rolled to one side or the other and the nose pitched down in order to prevent the temperature from falling.

The lower limits are designated as the "skip angle of attack" and the "skip bank angle". The reason for this terminology is that the spacecraft usually performs a nearly constant altitude glide at the beginning of Phase III, involving a very small skip, or increase in altitude, of between zero and 20,000 feet. For some situations, especially when the entry angle of attack is large, the skip does not occur; however, the designation is still retained.

Note that the skip angle of attack and the skip bank angle are controlled by the range guidance system. These attitude angles are always the actual values held by the vehicle whenever the temperature is falling. The skip angles may, of course, be changed at any time during Phase III, but an immediate change in predicted down range will not result if the skip angles are not in current use due to override by the CONTAC system.

The largest value used for the skip angle of attack was 55 degrees; the smallest value was 5 degrees. The smallest value might require adjustment if its use results in upper surface heating which cannot be tolerated.

The "largest" value used for the skip angle of roll was zero and the "smallest" value was -90 degrees. Notice that the use of a bank angle to the left preserves the algebraic definition of largest and smallest. If roll angles to the right are employed, the largest would still be zero and the "smallest" would then become +90 degrees. The key point to keep in mind is that the largest roll angle must always occur for the maximum vertical force (which obviously forces

the largest roll angle to be zero) and the "smallest" bank angle for the minimum vertical force.

The upper limits during Phase III are simply designated the "upper limits". The values are always the ones which give the maximum vertical force which is, of course, obtained at the angle for the maximum lift coefficient. Therefore, the upper limit for the angle of attack is 55 degrees and the upper limit for the bank angle is zero.

(d) Phase IV always starts with the attitude at the upper limits, the same as the upper limits for Phase III.

The lower limits for Phase IV are designated "lower limits". They are used when the speed falls appreciably and the CONTAC system goes into passive operation. Within the limitations imposed by the stability and available aerodynamic control of the vehicle, the guidance system is free to choose any values which it finds necessary to control the range. In general, the lower limit for the angle of attack will lie in the vicinity of the angle for $(L/D)_{max}$ while the lower limit for the bank angle is varied in the way which will result in the maximum cross range (if necessary).

(e) During Phase V, the CONTAC system is no longer in operation. All values for the attitude are dictated by guidance and control problems.

2.3 CONTAC Program Flow Diagram (see Fig. 12)

The basic purpose of the CONTAC system is to prevent a change in the temperature. The digital implementation of the basic concept is quite simple. The program to simulate and test the concept is, of course, much more detailed only because of other requirements such as program error checks and printouts of intermediate calculations. In general, conflicts often occur and the extra effort to take account of the numerous possibilities is considerable.

In the absence of these other details, the CONTAC system logic is shown in Fig. 12. There must be some memory (but no predictive requirement) so that the system can remember whether it is testing for a local minimum or maximum, and for the storage of $T_{last_{min}}$ or $T_{last_{max}}$, respectively. If the latter is currently being experienced, then when the temperature has fallen below the current local maximum by a reasonable margin ($T_{deadband}$), corrective action is taken. The determination of a local minimum is similar.

However the simulation of the CONTAC system by a digital computer is prone to difficulties in logic flow and numerical calculation which have no counterpart in the real situation. The basic cause is the method by which a digital computer must perform the calculations. As opposed to a continuous value for all variables, as is found in the output of an analogue computer, or analogue instruments, a digital computer calculates the flight path only at specific points along that path. As a consequence, the deadband test shown in Fig. 12 may not be satisfied at one point in the calculation, but will be satisfied by too great a margin at the next point. If the digital calculation is proceeding in steps of two or three seconds, the time lag introduced by the digital calculation is totally unacceptable (Fig. 13) and measures must be found

to correct the situation*. The simplest method would be to reduce the time interval (actually, the flight path distance) between the points in the calculation. However, what value should be used? A step size small enough to satisfy the deadband test requirement produces an unacceptably large increase in computing time on the machine. A step size which increases computer time by only a reasonable amount will not always be able to provide an acceptably short time delay.

The solution adopted was to provide the calculation section with backup capability. As shown in Fig. 14, new points are calculated at shorter and shorter intervals until both questions (Fig. 12) are satisfied. At that point, the distance travelled and the corresponding time delay between the actual position and the position at backup point No. 5 (Fig. 14) is acceptable and the commands to adjust the angle of attack and the bank angle are calculated and implemented.

The backup capability allows the calculation section to proceed with step sizes which are as large as calculation errors permit and thereby perform the calculation as fast as possible. The choice of the smallest step size for the test is based upon practical experience with the runs and usually results in a maximum time delay of between ten and twenty milliseconds. During test runs to determine the maximum acceptable value, it was found that delays which were an order of magnitude greater produced results which were only negligibly different. However, for still larger time delays, a discrepancy was noted sometimes.

2.4 Vehicle Implementation

The problems of isolating the control logic from the details of the calculation will, of course, disappear once the system is placed on board the spacecraft and actually put into operation. As shown in Fig. 15, the calculation section of Fig. 12 will be replaced by the vehicle dynamics. The vehicle sensors will provide the forward interface to the control hardware and the reaction jets will be the rearward interface. In the case of a digital control system, the sampling rates for the analogue to digital convertor will probably be on the order of ten milliseconds or less for the temperature. Other variables need to be sensed less often and perhaps only when needed.

The absence of inner return loops in the vehicle control logic merely indicates that no signal is necessary until the appropriate test is satisfied. However, since the calculation will be in real time, a great deal of attention must be given to the way in which the available computational time is divided. Because the CONTAC system is often in the passive mode, i.e. no active attitude control is being performed, time sharing procedures will probably make the best use of the available computational capacity. The range controller will undoubtably use the major share of the computer's time for predictive calculations since in view of the complex corrections required by the disturbing effects of the Earth's rotation and oblateness it is doubtful whether any simple predictive scheme would be satisfactory for the size of footprint within which the CONTAC system is able to operate. However, a simple interrupt system

---

* It is not so much the length of the time lag that complicates the situation, but the fact that it is unpredictable and varies from zero to perhaps five or even ten seconds. Since no real system will possess such a random variation in the size of the deadband, results computed on this basis would be highly suspect.

will allow for the monitoring of the temperature on a scheduled basis and the subsequent calculation of the attitude control moments when required.

The control scheme proposed here requires the measurement or computation of angles of attack and roll and their derivatives. No specific means are recommended for obtaining this information. However, one possibility is to compute them from the outputs of an intertial platform, which can provide both velocity and attitude in inertial space. Other techniques are also possible within the state of the art. When the CONTAC system commands a change of α or φ, it is assumed that pitch or roll jets are turned on, producing a constant moment with zero time lag. Both $|\dot{\alpha}|$ and $|\dot{\varphi}|$ are constrained to be less than certain selected maxima, and when these are reached, the control jets are turned off.

An extensive effort to produce a thermocouple which is capable of measuring the temperature at the nose stagnation point of the vehicle is described in Ref. 4. Actual tests were conducted which showed that such a device was able to measure values in excess of 3500°F for several minutes. Since the peak temperature which occurred for the vehicle used in this investigation was usually below 3000°F, there will not be any difficulty in procuring a satisfactory measuring instrument.

Another basic problem is to determine the actual stagnation point temperature. The position of the thermocouple is, of necessity, fixed in the nose of the vehicle. When changes are made in the angle of attack, the thermocouple will no longer be at the stagnation point and some method must be found to determine the actual stagnation point temperature.

The solution to this problem is also given in Ref. 4. The relationship between the measured and the stagnation point temperatures is given by

$$T_{measured} = T_{stagnation} \cos^{3/8} (\alpha - \alpha_{stagnation})$$

At high Mach numbers and for positions up to 40 degrees off the stagnation point, this relationship is in excellent agreement with both theoretical and experimental data for both hemispheres and swept cylinders. When the CONTAC system is active, the maximum variation in the angle of attack was from 5 degrees to 55 degrees. The thermocouple position could be at the stagnation point when the angle of attack of the vehicle is at 30 degrees. Consequently, the maximum offset would be only 25 degrees. For a stagnation point temperature of 3000°F, the temperature measured when the angle of attack of the vehicle is 55 degrees is only 4.2 percent less or 2875°F. The derivative of the temperature with respect to the angle of attack is also very small. The corresponding value would be 9.7 F° / degree change in angle of attack.

In general, the vehicle will experience a small, but variable degree of side slip, owing to the presence of inadvertent yawing movements from asymmetries and control action. Consequently, two extra thermocouples could be provided to measure the sideslip angle and to control reaction jets which would achieve a null angle. The suggested positions are on the wing leading edges; offset nose positions would also be satisfactory.

Note that regardless of the position used for the thermocouples, several extra sensors would be used in parallel at each location in order to guarantee reliability.

2.5 Vehicle Attitude Control

As part of any simulation of a complex physical situation, a decision must be made as to just how accurately the mathematical model should be able to duplicate the real situation. In the case of the attitude control system, information is available (Refs. 10 and 11) which makes possible a considerable simplification in the complexity of the mathematical model. These references indicate that the short period mode possesses time constants orders of magnitude greater than the characteristic times of the CONTAC system. As a result, most of the terms in the general equations describing the attitude of the vehicle may be eliminated - in particular, both the damping and spring constants are deemed negligible. Consequently, only the inertial term remains in the equations for rotation about any axis. It is also assumed that both the Earth's rotation and the movement of the vehicle within the Earth's reference system will have a negligible effect on the attitude equations.

In keeping with the above assumptions, the equations of motion for the attitude become very simple. Only the equation for the angle of attack will be developed since the bank angle equation is similar. The basic equation is

$$I_\alpha \ddot{\alpha} = M \tag{2.1}$$

where M is the constant pitching moment supplied by the pitch control jets when they are turned on. Thus

$$\ddot{\alpha} = k_\alpha \tag{2.2}$$

where k_α is zero when the control jets are off, positive when set for nose-up change, and negative for nose-down change.

Assuming that the angle of attack was decreasing when the pitch jets were first turned on and that the angle of attack should now begin to increase (i.e. k_α is positive), integration gives

$$\dot{\alpha} = \begin{cases} \dot{\alpha}_s + k_\alpha(t-t_s) & (t-t_s)<\Delta t \quad (2.3a) \\ \dot{\alpha}_{max} & (t-t_s)\geq\Delta t \quad (2.3b) \end{cases}$$

where

$$\Delta t = (\dot{\alpha}_{max} - \dot{\alpha}_s)/k_\alpha \tag{2.4}$$

A final integration yields

$$\alpha = \begin{cases} \alpha_s + \dot{\alpha}_s(t-t_s) + \frac{1}{2}k_\alpha(t-t_s)^2 & (t-t_s)<\Delta t \quad (2.5a) \\ \alpha_s + \Delta\alpha + \dot{\alpha}_{max}\left[(t-t_s)-\Delta t\right] & (t-t_s)\geq\Delta t \quad (2.5b) \end{cases}$$

where

$$\Delta\alpha = (\dot{\alpha}_{max} + \dot{\alpha}_s)(\dot{\alpha}_{max}-\dot{\alpha}_s)/2k_\alpha \tag{2.6}$$

Figure 16 shows the various relationships. Since equations (2.5a) and (2.5b) are simple analytical expressions which are a function only of time, they are easily introduced into the equations of motion presented in Section V. One point which may require explanation is the reason for the limit placed on $\dot{\alpha}$. On the one hand, as large a value as possible is desirable in order to be able to respond as quickly as possible to rapid changes in atmos-

pheric density that require a rapid adjustment of the angle of attack. On the other hand, if the inertia of the vehicle (with its negative angular velocity - remember that α_s was negative) is to be quickly overcome, Eq. (2.4) shows that a large $\dot{\alpha}_{max}$ requires a large angular acceleration. In addition, high angular rates lead to frequent cycling of the control which requires a high fuel expenditure for the reaction jets. Consequently, a reasonable compromise must be found between $\dot{\alpha}_{max}$, the available angular acceleration, and the available mass of fuel. The values used in this investigation were quite conservative. Appendix D gives a sample set of values prior to the actual output.

Slight variations in sign in Eqs. (2.3) to (2.6) are required when k_α is negative. Also, two sets of bank angle equations are required, one for positive k_φ and one for negative k_φ . The bank angle equations are a little more complicated to use since the optional right or left hand turn requires additional logic within the digital computer program. For a left hand turn, all the α symbols need only be replaced by φ to convert equations (2.3) to (2.6) into bank angle equations for a positive k_φ . Of course, the basic equation for the bank angle

$$\ddot{\varphi} = k_\varphi \tag{2.7}$$

which is exactly the parallel of Eq. (2.2) and applies equally to all situations.

III. VEHICLE CHARACTERISTICS

3.1 Introduction

In recent years, the general trend in lifting vehicle programs has been a shift away from configurations with relatively high lift to drag ratio ($L/D > 2.5$), such as the X-20 Dynasoar, to a lifting body style as typified by the M2-F2, the HL-10, and the SV-5 (Ref.12) wingless reentry vehicles. The reasons for the choice seem to have been based primarily upon the advantages associated with much lower temperatures resulting from the elimination of small nose cap and leading edge radii. The disadvantages caused by a much smaller footprint or landing area capability are not severe enough to cause serious concern. By reason of the Earth's rotation under what is essentially a fixed inertial orbit, these wingless vehicles are still able to land at any point on the Earth which falls underneath or at least near the orbital envelope (Fig. 5) without having to wait more than 24 hours. Naturally, the coverage is severely restricted for equatorial or near-equatorial orbits; but even the Dynasoar could not claim complete coverage for equatorial orbits without incurring extremely high temperatures.

3.2 Vehicle Geometry

The vehicle chosen for this study is quite similar to current design concepts. Its mass and wing area never individually enter the equations of motion; consequently, they have been left in the form of the wing loading ratio. This parameter was then chosen from about the middle of the range which is presently under review. The value of 40 lb/ft^2 is, therefore, simply representative.

The actual vehicle configuration is a delta wing planform with a leading edge sweep of about 70 degrees. Although the mass and wing area are not needed for the equations of motion, nevertheless some estimate of the nose cap and leading edge radii is desirable. A mass of about 8000 lb. gives a wing area of about 200 sq. ft. which would then lead to a base length of about 17 ft.

However, the body would be only about 20 ft. long since around 5 ft. of the nose would probably be cut off to provide a nose cap radius in the vicinity of 2.5 ft. The leading edge radius would also lie in this region.

3.3 Vehicle Aerodynamics

One of the most vital requirements for any investigation which seeks to predict the reentry flight path of a spacecraft with a reasonable degree of accuracy is a proper representation of aerodynamic forces throughout all phases of the trajectory. However, since the primary object of this study is to determine the effectiveness of the CONTAC system under varying operating conditions, only that portion of the flight where the heating rates are extreme will require highly accurate models for the aerodynamic forces which are exerted on the vehicle. Portions of the trajectory that are relatively unimportant to the primary purpose of the analysis may employ a model which is able to give only a reasonable approximation for the lift and drag coefficients.

Such is the case with the aerodynamic model chosen to represent the lift and drag forces on the vehicle under consideration. Several experimental studies (Refs. 13 to 15) show that expressions based on Newtonian theory adequately represent the sum total of the aerodynamic forces which are developed over a spacecraft during its reentry into the Earth's atmosphere as long as the Mach number is greater than about six or eight. As a result, a number of authors (Refs. 3, 16, and 17) have used either an exact or modified form of the flat plate, hypersonic, Newtonian approximation. Based on Ref. 14, the best estimate for the pressure coefficient on a flat plate is

$$C_p = 1.75 \sin^2\alpha \tag{3.1}$$

The value assigned to the minimum drag coefficient, $C_{D_{min}}$, is determined by the vehicle configuration, primarily by the degree of nose cap and leading edge bluntness. The nose cap radius should be as large as possible in order to reduce heating (See Sec. 3.4); however, a large value will produce a maximum lift-drag ratio which is too small to permit the required manoeuvrability. The compromise value of nose radius given in Sec. 3.2 yields a value for the minimum drag coefficient of

$$C_{D_{min}} = 0.0625 \tag{3.2}$$

If the value of the nose cap radius which has been suggested does not produce the above value for $C_{D_{min}}$, an adjustment to the former may easily be made without unduly affecting the peak temperature, since the latter varies only as the eighth root of the nose radius (see Eq. 3.7).

The use of the preceding values for C_p and $C_{D_{min}}$ gives the modified Newtonian expressions for the lift and drag coefficients, which are

$$C_L = 1.75 \sin^2\alpha \cos\alpha \tag{3.3}$$

$$C_D = 0.0625 + 1.69 \sin^3\alpha \tag{3.4}$$

These expressions are valid for $0 \leq \alpha \leq 90$ degrees and are quite accurate down to Mach numbers of six or seven. For Mach numbers smaller than five, there is every indication that Eqs. (3.3) and (3.4) continue to be quite representative even down through the transonic region. References 18 to 20 indicate that for

angles of attack below about twenty or thirty degrees, the lift and drag coefficients continue to exhibit the same general pattern while the L/D slowly increases with decreasing Mach number. Consequently, Eqs. (3.3) and (3.4) may continue in use down through the high subsonic range if conservative estimates of manoeuvring capability are acceptable.

3.4 Equilibrium Nose Stagnation Point Temperature

Since the primary goal of this investigation is to determine the applicability of the CONTAC system, the exact numerical magnitude of the nose stagnation point convective heating rate is relatively unimportant. Indeed, it is even questionable whether extremals in the stagnation point temperature (see Fig. 14 as to how a maximum or minimum T is handled) which occur as a result of reversals in the gradient of the heating rate must be accurately predicted by the model which has been adopted. On the other hand, the heating rate must be simulated in a way which is sensitive to changes in atmospheric and flight conditions, especially at the times when the model predicts that the maximum rate is being experienced. In other words, it is not at all important that the predicted heating rate reach a maximum or minimum at precisely the same time that the actual heating rate reaches its extremal values (in fact, it is very dubious whether any model which simulates the heating rate, no matter how complex or sophisticated it might be, could predict the exact time at which the heating rate derivative changes sign); it is only necessary that if the heating rate is to remain relatively constant as a result of attitude control for the simulated model that this condition also occur in the real situation and at about the same time in the flight. This rather lax requirement is due to the adaptive nature of the CONTAC system which must only prevent any large changes in the heating rate as long as possible, i.e. as long as the attitude angle has not reached its upper or lower limit.

The particular form of the nose stagnation point convective heating rate which is used in this investigation has also been adopted by several other authors (Refs. 3, 17, 21, and 22) as being sufficiently representative of experimental results. The basic relationship is

$$q_n = \frac{17,600}{\sqrt{R_n}} \left(\frac{V}{26,000}\right)^{3.15} \left(\frac{\rho}{0.002378}\right)^{0.5} \qquad (3.5)$$

Since the value which is to be measured on board the vehicle is the temperature, this heating rate is then inserted into the temperature rate equation which is

$$\frac{\bar{m}\,\bar{c}}{\bar{S}} \frac{d\hat{T}}{dt} = q - \epsilon\,\sigma\,\hat{T}^4 \qquad (3.6)$$

The radiation-equilibrium temperature is defined by setting the left hand side of Eq. (3.6) equal to zero. When an actual measurement is being made, the equilibrium value is produced by using such a small value of $\bar{m}\,\bar{c}/\bar{S}$ at the temperature sensor that the left hand side becomes negligible and an excellent approximation to the equilibrium value is achieved very quickly. The data in Ref. 4 indicates that there will be a relatively negligible time lag in reaching the equilibrium temperature since the time constants associated with the cyclic commands produced by the CONTAC system are of the order of five to ten seconds.

Then, setting to zero the left hand side of Eq. (3.6) and substituting Eq. (3.5) for the heating rate, q_n, gives the radiation-equilibrium nose

stagnation point temperature as

$$T^4 = \frac{17,600}{\epsilon\sigma\sqrt{R_n}} \left(\frac{V}{26,000}\right)^{3.15} \left(\frac{\rho}{0.002378}\right)^{0.5} \qquad (3.7)$$

The only terms which must have values assigned are ϵ and R_n. Normally, ϵ is between 0.5 and 0.8 while R_n is about 2.5 feet for the vehicle under consideration. Thus, the products of $\epsilon\sqrt{R_n}$ would range from 0.8 to 1.25 due to the uncertainty in ϵ. As a consequence, a "standard" value for the temperature is used for which a value of unity is taken for $\epsilon\sqrt{R_n}$. The advantage is a standard from which the proper result for the temperature may be computed when the correct values of ϵ and R_n are known without the disadvantage of being greatly in error. This standard stagnation point temperature is used in all the calculations as a measure of the actual temperature which would result for a device (e.g. a sophisticated thermocouple) placed at the nose stagnation point of the spacecraft. This variable will henceforth be referred to simply as the "temperature" (as already noted in Section 1.2). Then, in view of the above assumptions, the temperature is given by

$$T^4 = Q\, V^{3.15} \sqrt{\rho} \qquad (3.8)$$

where

$$Q = 17,600/\sigma 26,000^{3.15} \sqrt{0.002378} \qquad (3.9)$$

or

$$Q = 9,397 \qquad (3.10)$$

IV. GEOPHYSICAL ENVIRONMENT

4.1 Introduction

One of the most important goals of this investigation is to study the effect of non-ideal conditions upon the peak temperature, primarily with a view to determining the extent to which these conditions impair the performance of the CONTAC system.

There are two distinct types of non-ideal conditions which are of concern. The first includes all the effects which may be predicted well in advance of the reentry. These effects include the Earth's rotation, the Earth's oblateness (i.e. the distortions of both shape and gravitational potential), and the geographical variations of the time average density (which are a function of both latitude and longitude as well as altitude). The second type includes those effects which are the result of departures from the time averaged mean values of the density.

A third source of disturbance is atmospheric wind. No data could be found showing the spatial gradient of the wind at the altitudes of concern of this investigation (the same problem was encountered with density variations and will be discussed in Sec. 4.6). However, local wind velocities of up to 500 feet per second do exist. Actually, almost all values are lower than 250 feet per second, but on isolated occasions they approach the higher figure. Assuming a step increase of 500 feet per second (from -500 to +500 feet per second seems far too large a difference to support even if both conditions did occur within a reasonable proximity), the percentage increase in the velocity at a speed of 20,000 feet per second would amount to only $2\frac{1}{2}$ percent. The

resulting increase in temperature would be slightly less (see Eq. 3.8). The percentage increase would be greater at lower velocities, but by the time the speed falls below about 16,000 feet per second, the temperature is no longer a problem. Since it would be the changes in wind speed that cause the difficulties (see Sec. 8.4), it is clear that winds should not present a problem for the CONTAC system.

4.2 Rotation of the Earth

There is a relative pseudowind present due to the Earth's rotation. The West to East speed at the equator amounts to about 1500 feet per second relative to inertial space. For the calculation of aerodynamic heating and forces, this relative velocity may be subtracted from the initial entry speed for equatorial West to East reentries and provides a small but not insignificant reduction in the peak temperature experienced during the flight.

The actual rotational motion of the Earth is quite complex. Parvin (Reference 23) has done an excellent summary of the various motions and lists five separate parts to the rotation of the Earth.

The primary motion is, of course, the familiar rotation of the Earth about its axis which produces the normal sunrise and sunset. This movement takes by definition, exactly one solar day to complete. However, the rotational speed of interest is with respect to inertial space and for this figure the sideral day may be used. The length of time taken for one revolution in inertial space is 23 hours, 56 minutes, and 4.09 seconds. This works out to a rotational speed of 0.00417807 degrees of longitude per second.

The other four parts of the rotation will be mentioned for interest, but are all negligible with regard to the effect they have on the reentry of a spacecraft First, there is the precessional rotation about the 23.5 degree inclination to the ecliptic pole which takes about 25,800 years. Second, other bodies in the solar system tend to cause a slight irregularity in the precessional rate called nutation; one component with less than ten seconds of arc has a period of nineteen years while the other component with less than one-half second of arc has a period of only two weeks. Next, the pole itself wanders, which would tend to affect the alignment of the Earth's axis; however, although the two periods are relatively short - twelve and fourteen months - the angles are less than four seconds of arc. Finally, there is an additional revolution due to the spiral motion of the galaxy which occurs once approximately every 200 million years. A better and much more complete description of these four additional motions may be found in the reference. It is quite clear, however, that they are too small to have any effect on the reentry.

4.3 Earth's Oblate Gravitational Field

During the past ten years, the use of artificial satellites to measure the potential field of the Earth has shown the inadequacy of representing the gravitational attraction as a simple central force. Using the notation of this paper, the general representation of the Earth's potential field as given by Kozai (Ref. 24) is

$$U = \frac{GM}{r}\left[1 + \sum_{m=2}^{\infty}\sum_{m=0}^{\infty}\left(\frac{R_o}{r}\right)^n P_n^m(\sin\lambda)\{C_{n,m}\cos m\theta + S_{n,m}\sin m\theta\}\right] \quad (4.1)$$

where

$$P_n^m(\sin\lambda) = \frac{1}{2^n n!}\cos^m\lambda \frac{d^{n+m}(\sin^2\lambda - 1)^n}{d(\sin\lambda)^{n+m}} \quad (4.2)$$

are spherical hormonics. Several other formulations exist, but all are essentially the same in that they use an infinite series of spherical harmonics to represent the potential field.

The gravitational vector is divided into its three principal components which results in (see Sec. 5.2 for a definition of the $\{\bar{n},\bar{e},\bar{v}\}$ reference system)

$$\bar{g} = \bar{n}\, g_\lambda + \bar{e}\, g_\theta - \bar{v}\, g_r \quad (4.3)$$

and requires the expressions for each of the gravitational components which are defined by the spherical gradient of the potential as

$$g_r = \frac{\partial U}{\partial r} \quad (4.4a)$$

$$g_\lambda = \frac{1}{r}\frac{\partial U}{\partial \lambda} \quad (4.4b)$$

$$g_\theta = \frac{1}{r\cos\lambda}\frac{\partial U}{\partial \theta} \quad (4.4c)$$

Substitution of U into equations (4.4a), (4.4b), and (4.4c) and expansion of the spherical harmonics by means of Eq. (4.2) is tedious, but can easily be carried out. The result for the first few terms in the series is shown in order to provide the reader with some idea of the actual results. When all the substitutions are made, Eq. (4.3) for $\bar{g}$ becomes

$$\begin{aligned}\bar{g} = -\bar{v}\,\frac{GM}{r^2}\Bigg[1 &+ \left(\frac{R_o}{r}\right)^2\left\{\frac{3}{2}(3\sin^2\lambda - 1)C_{2,0} + 9\cos^2\lambda(C_{2,2}\cos 2\theta + S_{2,2}\sin 2\theta)\right\} \\ &+ \left(\frac{R_o}{r}\right)^3\Big\{2(5\sin^3\lambda - 3\sin\lambda)C_{3,0} \\ &\quad + 2(1.5\sin^2\lambda - 3)\cos\lambda(C_{3,1}\cos\theta + S_{3,1}\sin\theta) \\ &\quad + 60\sin\lambda\cos^2\lambda(C_{3,2}\cos 2\theta + S_{3,2}\sin 2\theta) \\ &\quad + 60\cos^3\lambda(C_{3,3}\cos 3\theta + S_{3,3}\sin 3\theta)\Big\} + \ldots.\Bigg]\end{aligned}$$

$$+ \bar{n}\ \frac{GM}{r^2}\left[\left(\frac{R_o}{r}\right)^2 \left\{3\ \sin\lambda\cos\lambda C_{2,0} - 6\ \sin\lambda\cos\lambda(C_{2,2}\cos2\theta + S_{2,2}\sin2\theta)\right\}\right.$$

$$+ \left(\frac{R_o}{r}\right)^3 \left\{ \tfrac{1}{2}(15\ \sin^2\lambda - 3)\cos\lambda C_{3,0} \right.$$

$$+ \tfrac{3}{2}(11 - 15\ \sin^2\lambda)\sin\lambda(C_{3,1}\cos\theta + S_{3,1}\sin\theta)$$

$$+ 15(1 - 3\ \sin^2\lambda)\cos\lambda(C_{3,2}\cos2\theta + S_{3,2}\sin2\theta)$$

$$\left.\left. + 45(\sin^2\lambda - 1)\sin\lambda(C_{3,3}\cos3\theta + S_{3,3}\sin3\theta)\right\} + \ldots\right]$$

$$+ \bar{e}\ \frac{GM}{r^2}\left[\left(\frac{R_o}{r}\right)^2 \left\{6\cos\lambda(S_{2,2}\cos2\theta - C_{2,2}\sin2\theta)\right\}\right.$$

$$+ \left(\frac{R_o}{r}\right)^3 \left\{ \tfrac{1}{2}\ (15\ \sin^2\lambda - 3)(S_{3,1}\cos\theta - C_{3,1}\sin\theta) \right.$$

$$+ 30\ \sin\lambda\cos\lambda(S_{2,2}\cos2\theta - C_{2,2}\sin2\theta)$$

$$\left.\left. + 45(1 - \sin^2\lambda)(S_{3,3}\cos3\theta - C_{3,3}\sin3\theta)\right\} + \ldots\right] \tag{4.5}$$

Although the forces due to even higher order harmonics will influence the orbit of an artificial satellite to some extent, the presence of terms other than those containing $C_{2,0}$ has a negligible effect upon the reentry of a spacecraft. By neglecting all such terms, the complexity of the gravitational vector is considerably reduced, with the $\{\bar{e}\}$ component disappearing altogether. The result is

$$\bar{g} = \bar{v}\ \frac{GM}{r^2}\left[1 + \left(\frac{R_o}{r}\right)^2 \frac{3}{2}(3\ \sin^2\lambda - 1)C_{2,0}\right]$$

$$+ \bar{n}\ \frac{GM}{r^2}\left[\left(\frac{R_o}{r}\right)^2 3\ \sin\lambda\cos\lambda C_{2,0}\right] \tag{4.6}$$

The three remaining constants are given by

$$GM = 3.986032 \times 10^{20}\ \text{cm}^3/\text{sec}^2 \tag{4.7}$$

or

$$GM = 1.407654 \times 10^{16}\ \text{ft}^3/\text{sec}^2$$

$$R_o = 6.378196 \times 10^8\ \text{cm} \tag{4.8}$$

or

$$R_o = 20{,}925{,}840\ \text{feet}$$

$$C_{2,0} = -1.08248 \times 10^{-3} \tag{4.9}$$

They were taken as the best average of all the values suggested by Refs. 24 to 26.

4.4 Oblate Shape of the Earth

The primary departure of the Earth from a perfect sphere is the flattening of the poles. Several papers by Kaula (Refs. 25 and 26 in particular) have also indicated the additional distances by which the Earth departs from its ellipsoidal shape, which in turn gives rise to the additional coefficients in Eq . (4.1). The variations are systematic to the extent that they indicate a triaxial tendency. Wagner (Refs. 27 to 29) has done extensive work on this subject with, however, conflicting results. Consequently, only an approximate (one figure accuracy) average was adopted with the major axis at about twenty degrees longitude west and a minor axis about 210 feet shorter ($f_q = 10^{-5}$) situated at seventy degrees longitude east.

In order to calculate the perpendicular height above a triaxial ellipsoid, it is sufficient to know the geocentric geographical position. The three variables will be the geocentric radius, the geocentric latitude, and the geocentric longitude. Several different methods were found in the literature which provided suggestions on how to calculate the perpendicular height above an oblate spheroid. While most were unnecessarily complex, a very simple solution is given by Baker (Ref. 30) which is deemed to be the best. In the notation of this paper, the suggested equation is

$$y = r - R_o \left[1 - f_p \sin^2\lambda - \tfrac{1}{2} (f_p)^2 \left(\frac{R_o}{r} - \tfrac{1}{4} \right) \sin^2 2\lambda \right] \qquad (4.10)$$

where y is the perpendicular height above an ellipsoidal Earth.

In order to compensate for the triaxial tendency of the Earth's equator, an additional term of the form $R_o f_q \sin^2\theta$ (which is derived in the same way as the $R_o f_p \sin^2\lambda$ term for the polar flattening) is required. The higher order term in f_q^2 is, of course, negligible in comparison. This additional term must then be multiplied by $\cos^2\lambda$ in order to reduce the triaxial effect to zero at the poles. A biasing constant is also required in order to raise the major axis and lower the minor axis above and below, respectively, the Earth's average equatorial radius, R_o. This biasing constant(of $\frac{1}{2}$) in equation (4.11) is required due to the definition of f_q as given in the nomenclature.

In addition, $\sin^2 2\lambda$ is expanded using the usual trigonometric relationship in order to adapt the formula for digital computation (both $\sin\lambda$ and $\cos\lambda$ are already available, but $\sin 2\lambda$ would have to be calculated). The result is

$$\begin{aligned} y = r - R_o + R_o f_p \sin^2\lambda + (f_r/r + f_c)\sin^2\lambda\cos^2\lambda \\ + R_o f_q \{\sin^2(\theta - \theta_{2,2}) - \tfrac{1}{2}\}\cos^2\lambda \end{aligned} \qquad (4.11)$$

An exact procedure which required several iterations to yield the result and which was not, therefore, useful during the actual computation of trajectories was then compared with the answers from Eq. (4.11). The exact procedure indicated that a correction factor should be added to take account of higher order terms which were omitted from the power series approximation given in Eq. (4.10). The actual formula used during calculation of trajectories was

$$\begin{aligned} y = r - R_o + R_o f_p \sin^2\lambda + \{f_r/r + f_c + f_f(\sin^2\lambda - \tfrac{1}{2})\} \sin^2\lambda\cos^2\lambda \\ + R_o f_q \{\sin^2(\theta - \theta_{2,2}) - \tfrac{1}{2}\} \cos^2\lambda \end{aligned} \qquad (4.12)$$

where

$$R_o = 6.378196 \times 10^8 \text{cm}$$

or

$$R_o = 20{,}925{,}840 \text{ feet} \tag{4.13}$$

$$f_p = 1/298.28 \tag{4.14}$$

$$f_q = 10^{-5} \quad \theta_{2,2} = -20 \text{ degrees} \tag{4.15}$$

were taken as the best average of all the values suggested by Refs. 24 to 29.

Also

$$f_r = \tfrac{1}{2}(R_o f_p)^2 \tag{4.16}$$

$$f_c = R_o(f_p)^{2/8} \tag{4.17}$$

And

$$f_f = -3.328 \tag{4.18}$$

was optimized for the adopted values of R_o and f_p.

In practice, Eq. (4.12) evaluated the last three terms in single precision arithmetic and then used double precision addition when adding all five values together to achieve the final result. Comparison with the exact iterative procedure showed that the accuracy achieved with Eq. (4.12) was always better than 0.05 feet over the tested altitude range of zero to one million feet.

4.5 Mean Standard Atmosphere

In this investigation, the basic model used to define the variation of density with altitude was the U.S. Standard Atmosphere, 1962 (Ref. 31). As far as corrections for oblateness were concerned, the true perpendicular altitude above the surface was used to define the height as per Eq. (4.12). In essence, the atmosphere was assumed to be a uniformly thick blanket enveloping the Earth. Figure 17 from a report by Cole and Kantor (Ref. 32) gives ample evidence to support his assumption.

Additional information regarding standard atmospheres for different latitudes has recently appeared in the literature (Ref. 33). However, in view of the lack of detail concerning the space between each mean set of values, the additional accuracy afforded by the extra reference curves would not have been worth the additional programming effort necessary to make them available to the program calculating the flight trajectories. In addition, there must certainly be longitudinal variations, as well as those associated with latitude, and the current absence of information about these is further justification for ignoring both effects. Perhaps the most important point in favour of using a single standard atmospheric model over the whole Earth is the necessity of determining the effects of departures from a non-rotating, circular Earth in a manner which distinguishes them from those due to atmospheric density variations.

4.6 Atmospheric Density Variations

The presence of spatial density variations about some mean values in the atmosphere poses a problem with which any control system must cope. Whether the purpose of the control system is to achieve lower temperatures, a terminal position, or any other of several possibilities, the most difficult part of the problem is the fact that the density variations about the mean are random and can only be specified by way of statistical data. Since this paper deals with the feasibility of the CONTAC system rather than with the exact details of the hardware, a single mean curve, the U.S. Atmosphere, 1962, was deemed sufficiently representative for all but a few reentries. Then, the problem was to find a model that would permit a reasonably accurate picture of the spatial density variations to be found in the atmosphere about the mean curve of density versus altitude.

One question that was deemed important, however, was whether the shape of the mean curve of density versus altitude was itself of any great importance. In order to answer it, a number of quite arbitrary exponential curves were used with widely varying values of the scale height. Actually, each new reference curve was composed of two exponentials. The higher altitude portion was defined by the scale height and the point at which it intersected the U.S. Standard Atmosphere, 1962. The lower curve was defined by the cut-off point for the upper curve and a second point of intersection with the U.S. Standard Atmosphere, 1962 which was below the cut-off point. The results are given in Sec. 8.2.

Two types of spatial density variations about the mean were used. One was based on altitude and the other was generated as a set of Gaussian random numbers which were spaced at equal intervals along the flight path length. Considerable attention was paid to the choice of these two types due to the almost complete lack of data regarding spatial density variations in the atmosphere. A great many papers could be found which gave day to day variations at a given point. Other investigators gave the results of calculations which defined new mean atmospheres at various latitudes. However, Cole and Cantor (Ref. 32) did provide some information regarding the maximum spatial density variations about the mean which might be expected as well as the distances over which the departures might occur.

For the density variations based on altitude, an actual experimental result shown in Ref. 32 confirmed a suggestion by Etkin (Ref. 34) that a simple sinusoidal variation about the mean of density versus altitude would be realistic. The necessary parameters were the amplitude of the variation about the mean, the wave length, and the point of zero variation or the nodal position. A range of values were used and the results can be found in Sec. 8.3.

In the case of random departures from the basic curve, a statistical approach was required. The method chosen to simulate the variations about the mean was a series of random numbers representing fractional deviations from the mean, with a Gaussian distribution. After specifying the variance of the series, the numbers were then assigned to equidistance positions along the flight path. In the absence of any other data, linear interpolation was used between the points to find the fractional departure from the mean value for the density. The fraction could be positive or negative, but was limited to an absolute magnitude of three times the variance (which it would be less than 99%

of the time in any case) in order to prevent bizarre results such as a negative density. In order to provide a statistical result, a number of trajectories was computed, one for each of several different series of random numbers having the same variance and correlation interval. (The effect of the random density model is presented in Sec. 8.4.)

While the above method of simulating the random variation to be expected in the atmosphere is certainly not the best possible model, it should at the very least be adequate. In any case, the lack of any concrete data upon which to base a more realistic model does not warrant a more sophisticated approach. It is, of course, possible that the information exists in classified reports. If so, the results presented in Sec. 8.4 will provide an immediate answer as to whether the CONTAC system will perform in a satisfactory manner under all expected atmospheric variations. On the other hand, if such data actually does not exist, then an evaluation of the CONTAC system must wait until it has been determined.

V. EQUATIONS OF MOTION

5.1 Basic Reference System

The choice of a basic reference system from among the multitude of possibilities is usually dictated by the problem which is to be solved as well as by the methods of solution which are available. In many situations, as in the present case, the choice is easily made due to the many restrictions imposed by urfavourable reference systems which might require an unnatural set of equations or perhaps extra terms in the equations. Initially, this investigation was begun using an analogue computer of limited capacity. When the problem outgrew the available equipment, a large scale digital computer was substituted. The original reference system was retained, however, due to the background of experience and confidence.

The basic reference system chosen is a set of axes $\{\bar{I},\bar{J},\bar{K}\}$ fixed at the mass centre of a rotating Earth (Fig. 18). Thus, the position of the vehicle is naturally expressed in terms of the geocentric latitude, λ, the geocentric longitude, θ(positive to the East), and the geocentric radius, r (Fig. 19); the velocity of the vehicle is then given by the geocentric azimuth, ψ (heading with respect to the North-positive to the East), the glide angle, γ (angle between the local geocentric horizontal and the velocity vector-positive downward), and the magnitude of the velocity vector, $\bar{V}$ (Fig. 19). However, since all aerodynamic data is always referenced to a set of axes which are aligned in some way with the vehicle, a second reference system which will be termed flight axes $\{\bar{i},\bar{j},\bar{k}\}$ must be introduced (Fig. 20). It should be noted that these flight axes are not fixed to the body; on the contrary, the $\{\bar{j}\}$ axis always remains in the local geocentric horizontal plane, thereby forcing the $\{\bar{I},\bar{k}\}$ axes to remain in the local geocentric vertical plane. In order to complete the orientation requirements, the $\{\bar{i}\}$ axis is directed along the velocity vector, $\bar{V}$, and the $\{\bar{k}\}$ axis is directed downward (in this study $|\gamma| < 90$ degrees). As a consequence of this definition for the flight axes, the side force equation will include the component of the lift in that direction, namely: $L \sin \varphi$. For those readers who have not encountered this vehicle reference system of flight axes (for example, Refs. 35 and 36), it should again be mentioned that this investigation is concerned with a three degree of freedom system of equations

for a point mass vehicle. Since the pitch, roll, and yaw angles which are dependent variables in a six degree of freedom system are now control variables, it is rather useful to simplify the three force equations by using the required components of the lift and drag. The point mass vehicle is then rotated in pitch and roll (sideslip is assumed negligible and the yaw angle is, therefore, set identically equal to zero) so as to develop these required lift and drag components. The validity of the assumption regarding the representation of aerodynamic forces has been examined in Section III.

5.2 Conversion Between Reference Systems

Since the equations of motion are written in the flight axes reference system, it is necessary to find the components of all forces in this final reference system. (Fig. 21). Thus the conversion matrices from any of the basic or intermediate reference systems to the flight axes reference system must be found. A rotation about the $\{\bar{K}\}$ axis (Fig. 22) gives the transformation from the $\{\bar{I},\bar{J},\bar{K}\}$ set of axes to the $\{\bar{I}',\bar{J}',\bar{K}\}$ set of axes. The second rotation about the $\{\bar{J}'\}$ axis (Fig. 23) gives the transformation to the $\{\bar{n},\bar{e},\bar{v}\}$ set of axes. The third rotation about the $\{\bar{v}\}$ axis (Fig. 24) gives the transformation to the $\{\bar{h},\bar{J},\bar{v}\}$ set of axes. Finally, the fourth rotation about the $\{\bar{j}\}$ axis gives the transformation to the $\{\bar{i},\bar{j},\bar{k}\}$ reference system of flight axes (Fig. 25). These four rotations naturally require the four simple conversion matrices given with each figure. The overall conversion matrices from any of the intermediate reference systems are then easily found to be

$$\begin{Bmatrix}\bar{h}\\ \bar{j}\\ \bar{v}\end{Bmatrix} = \begin{pmatrix}\cos\gamma & 0 & -\sin\gamma\\ 0 & 1 & 0\\ \sin\gamma & 0 & \cos\gamma\end{pmatrix}\begin{Bmatrix}\bar{i}\\ \bar{j}\\ \bar{k}\end{Bmatrix} \tag{5.1}$$

$$\begin{Bmatrix}\bar{n}\\ \bar{e}\\ \bar{v}\end{Bmatrix} = \begin{pmatrix}\cos\gamma\cos\psi & -\sin\psi & -\sin\gamma\cos\psi\\ \cos\gamma\sin\psi & \cos\psi & -\sin\gamma\sin\psi\\ \sin\gamma & 0 & \cos\gamma\end{pmatrix}\begin{Bmatrix}\bar{i}\\ \bar{j}\\ \bar{k}\end{Bmatrix} \tag{5.2}$$

$$\begin{Bmatrix}\bar{I}'\\ \bar{J}'\\ \bar{K}\end{Bmatrix} = \begin{pmatrix}\cos\gamma\cos\psi\sin\lambda & -\sin\psi\sin\lambda & -\sin\gamma\cos\psi\sin\lambda\\ +\sin\gamma\cos\lambda & & +\cos\gamma\cos\lambda\\ -\cos\gamma\sin\psi & -\cos\psi & \sin\gamma\sin\psi\\ \cos\gamma\cos\psi\cos\lambda & -\sin\psi\cos\lambda & -\sin\gamma\cos\psi\cos\lambda\\ -\sin\gamma\sin\lambda & & -\cos\gamma\sin\lambda\end{pmatrix}\begin{Bmatrix}\bar{i}\\ \bar{j}\\ \bar{k}\end{Bmatrix} \tag{5.3}$$

The conversion matrix between the basic reference system $\{\bar{I},\bar{J},\bar{K}\}$ and the flight axes reference system $\{\bar{i},\bar{j},\bar{k}\}$ could be found and expressed in the same manner; however, it is unnecessary since it is not used.

5.3 Vector Equation of Motion

An evaluation was made of the relative magnitudes of all forces which may act upon a spacecraft during a reentry into the Earth's atmosphere. Forces due to other planetary bodies including the moon, the solar wind, and magnetic fields often have an appreciable effect on the orbits of particular satellites when these forces are experienced over long periods of time. However,

for a reentry trajectory, it can easily be shown (Ref. 38) that only aerodynamic forces and the Earth's gravitational field have a non-negligible effect upon the flight path. In addition, based on the information in Ref. 23, it may be assumed that the basic reference system, $\{\bar{I},\bar{J},\bar{K}\}$ (see Fig. 18), is rotating in inertial space at a constant angular velocity.

The single vector equation of motion which is required for a point mass moving in such a rotating reference system is given by (Ref. 39)

$$\frac{\bar{F}}{m} = \bar{a} + 2(\bar{\omega}x\bar{V}) + \bar{\omega}\, x(\bar{\omega}x\bar{r}) \tag{5.4}$$

where $\bar{a}$ is the acceleration relative to the $\{\bar{I},\bar{J},\bar{K}\}$ reference system, i.e. relative to the Earth.

The only external forces are either aerodynamic or gravitational. It is appropriate, therefore, to separate the left hand side of Eq. (5.4) and write

$$\bar{F} = \bar{F}_A + m\bar{g} \tag{5.5}$$

The relative acceleration, $\bar{a}$, in the $\{\bar{I},\bar{J},\bar{K}\}$ system has components along the $\{\bar{i},\bar{j},\bar{k}\}$ axes that are obtained by the rule for transforming a derivative between rotating frames of reference (Ref. 40), i.e.

$$\bar{a} = \dot{\bar{V}} + (\bar{\Omega}x\bar{V}) \tag{5.6}$$

where $\bar{\Omega}$ is the angular velocity of the $\{\bar{i},\bar{j},\bar{k}\}$ system relative to the $\{\bar{I},\bar{J},\bar{K}\}$ and the components of $\bar{V}$ are the rates of change of the components of $\bar{V}$ along the the $\{\bar{i},\bar{j},\bar{k}\}$ axes.

Equations (5.4) to (5.6) can be combined to give

$$\frac{\bar{F}_A}{m} + \bar{g} = \dot{\bar{V}} + (\bar{\Omega}x\bar{V}) + 2(\bar{\omega}x\bar{V}) + \bar{\omega}x(\bar{\omega}x\bar{V}) \tag{5.7}$$

5.4 Expansion of the Single Vector Equation of Motion

The expansion of each of the terms in Eq. (5.7) in the $\{\bar{i},\bar{j},\bar{k}\}$ flight axes system is not difficult although it is somewhat involved and lengthy. The three components of each term will first be found in whichever axes system is most convenient then, if necessary, transferred to the $\{\bar{i},\bar{j},\bar{k}\}$ flight axes system, and finally collected to give the three complete force equations. Starting with the first term, the results are as follows.

The aerodynamic forces are given by

$$\begin{aligned}\bar{F}_A = &- \bar{I}\, C_D(\alpha)\, \tfrac{1}{2}\rho(y)\, V^2\, S \\ &+ \bar{j}\, C_L(\alpha)\, \tfrac{1}{2}\rho(y)\, V^2\, S \sin\varphi \\ &-\bar{k}\, C_L(\alpha)\, \tfrac{1}{2}\rho(y)\, V^2\, S \cos\varphi\end{aligned} \tag{5.8}$$

$C_L(\alpha)$ and $C_D(\alpha)$ are given by Eqs. (3.3) and (3.4). The expression for the density, $\rho(y)$, has been discussed in Sec. 4.5 and the evaluation of the height y, above a triaxial ellipse is given by Eq. (4.12).

The gravitational force vector is given by Eq. (4.3) in the $\{\bar{h},\bar{e},\bar{v}\}$ reference system. Transforming the components from this intermediate reference system to the $\{\bar{i},\bar{j},\bar{k}\}$ flight axes system via equation (5.2) gives the gravity vector as

$$\begin{aligned}\bar{g} = {} & \bar{i}\,(-g_r \sin\gamma + g_\lambda \cos\gamma\cos\psi + g_\theta \cos\gamma\sin\psi) \\ & + \bar{j}\,(-g_\lambda \sin\psi + g_\theta\cos\psi) \\ & + \bar{k}\,(-g_r \cos\gamma - g_\lambda \sin\gamma\cos\psi - g_\theta\sin\gamma\sin\psi)\end{aligned} \tag{5.9}$$

The expressions for g_r, g_λ, and g_θ are developed and stated in Sec. 4.3. Although, as noted in Sec. 4.3, g_θ is too small to affect the reentry trajectory of a spacecraft, the expressions have purposely been left in a generalized form so that the precise effect of any term in the series may be ascertained should it be desirable at some future date. In addition, the values of g_r and g_λ will normally include only the major effect of the Earth's oblate gravity potential, namely $C_{2,0}$ term. However, here again, the expressions are left in their generalized form. The truncated expressions which were actually used in the calculations may be found by a comparison of Eqs. (4.3) and (4.6) to be

$$g_r = -\frac{GM}{r^2}\left[1 + \left(\frac{R_o}{r}\right)^2 \frac{3}{2}\,(3\sin^2\lambda - 1)C_{2,0}\right] \tag{5.10}$$

$$g_\lambda = \frac{GM}{r^2}\left[\left(\frac{R_o}{r}\right)^2 3\sin\lambda\cos\lambda C_{2,0}\right] \tag{5.11}$$

$$g_\theta = 0 \tag{5.12}$$

The terms on the right hand side of Eq. (5.7) are either very simple or quite complex. The first term is simply

$$\dot{\bar{V}} = \bar{i}\,\dot{V} \tag{5.13}$$

Before proceeding with the next term, the expression for $\bar{\Omega}$ should be found. With reference to Figs. 22 to 25, it can easily be seen that the sum total of all the rotations which are required to rotate from the $\{\bar{I},\bar{J},\bar{K}\}$ basic reference system to the $\{\bar{i},\bar{j},\bar{k}\}$ flight axes reference system yields the relative rotational velocity which is

$$\bar{\Omega} = \bar{K}\,\dot{\theta} + \bar{J}'\,\dot{\lambda} + \bar{v}\,\dot{\psi} - \bar{j}\,\dot{\gamma} \tag{5.14}$$

The negative sign in the last term is required due to the rotation of $\bar{h}$ into $\bar{v}$ which is in the opposite sense from the normal right hand rule.

At this point, the three kinematic relationships for the position of the vehicle $\dot{r}$, $\dot{\lambda}$ and $\dot{\theta}$ should be defined since the last two are required immediately. They are

$$\dot{r} = -V\sin\gamma \tag{5.15}$$

$$\dot{\lambda} = \frac{V}{r}\cos\gamma\cos\psi \tag{5.16}$$

$$\dot{\theta} = \frac{V\cos\gamma\sin\psi}{r\cos\lambda} \tag{5.17}$$

Then, in order to transform $\bar{\Omega}$ into the $\{\bar{i},\bar{j},\bar{k}\}$ flight axes system, the trans-

formation given in Eqs. (5.2) and (5.3) must be used, along with the substitution of Eqs. (5.16) and (5.17). The result is

$$\bar{\Omega} = \bar{i}(\dot{\psi}\sin\gamma - \frac{V}{r\cos\lambda}\sin\gamma\cos\gamma\sin\lambda\sin\psi) + \bar{j}(-\dot{\gamma} - \frac{V}{r}\cos\gamma) + \bar{k}(\dot{\psi}\cos\gamma - \frac{V}{r\cos\lambda}\cos^2\gamma\sin\lambda\sin\psi) \quad (5.18)$$

Finally, combining the expression for $\bar{\Omega}$ with $\bar{V} = \bar{i}\,V$ yields the desired cross product for the second term on the right hand side of Eq. (5.7). The result is

$$(\bar{\Omega}\times\bar{V}) = \bar{i}\,(0) + \bar{j}\,(\dot{\psi}V\cos\gamma - \frac{V^2}{r\cos\lambda}\cos^2\gamma\sin\lambda\sin\psi) + \bar{k}\,(\dot{\gamma}\,V + \frac{V^2}{r}\cos\gamma) \quad (5.19)$$

The next term is somewhat easier to express. It is easily seen (Fig. 23) that

$$\bar{\omega} = \bar{K}\,\omega \quad (5.20)$$

which upon substitution of Eq. (5.3) and combination with $\bar{V} = \bar{i}\,v$ results in

$$2(\bar{\omega}\times\bar{V}) = \bar{i}\,(0) - \bar{j}(2\omega V[\sin\gamma\cos\lambda\cos\psi + \cos\gamma\sin\lambda]) + \bar{k}\,(2\omega V\cos\lambda\sin\psi) \quad (5.21)$$

Finally, the last term is easiest to expand in the reverse direction. It is also easy to see (Fig. 23) that

$$\bar{r} = -\bar{v}\,r \quad (5.22)$$

or

$$\bar{r} = \bar{I}'r\cos\lambda - \bar{K}\,r\sin\lambda \quad (5.23)$$

which upon combination with Eq. (5.20) gives

$$\bar{\omega}\times(\bar{\omega}\times\bar{r}) = \bar{I}'\,\omega^2\,r\cos\lambda \quad (5.24)$$

and after substitution of Eq. (5.3) results in

$$\bar{\omega}\times(\bar{\omega}\times\bar{r}) = \bar{i}\,(\omega^2 r\cos\lambda[\sin\gamma\cos\lambda + \cos\gamma\sin\lambda\cos\psi]) - \bar{j}\,(\omega^2 r\cos\lambda\sin\lambda\sin\psi) + \bar{k}\,(\omega^2 r\cos\lambda[\cos\gamma\sin\lambda - \sin\gamma\sin\lambda\cos\psi]) \quad (5.25)$$

All of the terms in Eq. (5.7) have now been expanded into the $\{\bar{i},\bar{j},\bar{k}\}$ flight axes reference system. Since each of the three components are

independent of the other, three first order, non-linear, differential equations may be written by collection of all the terms in Eqs. (5.8), (5.9), (5.13), (5.19), (5.21), and (5.25). The result is

$$\dot{V} = - C_D(\alpha)\tfrac{1}{2}\rho(y)V^2S/m - g_r\sin\gamma + g_\lambda\cos\gamma\cos\psi + g_\theta\cos\gamma\sin\psi$$
$$-\omega^2 r\cos\lambda(\sin\gamma\cos\lambda + \cos\gamma\sin\lambda\cos\psi) \qquad (5.26)$$

$$\dot{\gamma}V = - C_L(\alpha)\tfrac{1}{2}\rho(y)V^2\frac{S}{m}\cos\varphi - \frac{V^2}{r}\cos\gamma - g_r\cos\gamma - g_\lambda\sin\gamma\cos\psi$$
$$- g_\theta\sin\gamma\sin\psi - \omega^2 r\cos\lambda(\cos\gamma\cos\lambda - \sin\gamma\sin\gamma\cos\psi)$$
$$- 2\omega V\cos\lambda\sin\psi \qquad (5.27)$$

$$\dot{\psi}V\cos\gamma = C_L(\alpha)\tfrac{1}{2}\rho(y)V^2\frac{S}{m}\sin\varphi + \frac{V^2\cos^2\gamma\sin\lambda\sin\psi}{r\cos\lambda} - g_\lambda\sin\psi + g_\theta\cos\psi$$
$$+ \omega^2 r\cos\lambda\sin\lambda\sin\psi + 2\omega V(\sin\gamma\cos\lambda\cos\psi + \cos\gamma\sin\lambda) \qquad (5.28)$$

In addition to the above three differential equations, and the positional differential equations (5.15), (5.16), and (5.17), several other relationships are required which have already been stated or developed previously. For convenience, they will be mentioned again at this point. Equations (3.3) and (3.4) give the relationship between the lift and drag coefficients, $C_L(\alpha)$ and $C_D(\alpha)$, and the angle of attack, α. The relationship between the density, $\rho(y)$, and the perpendicular height above the Earth's surface, y, has been discussed in Sections 4.5 and 4.6. The perpendicular height above the surface of a triaxial Earth, y, is given by Eqs. (4.12) to (4.18). Finally, the expressions for g_r, g_λ, and g_θ are given by Eqs. (5.10), (5.11), and (5.12). Should it be desirable to include additional terms in the oblate gravity field of the Earth, equations (4.1) to (4.5) will serve to illustrate the manner by which it may be done.

At this point in the development of the equations of motion, it was found to be convenient to make a change in the independent variable from time, t, to path length along the trajectory, s. The original reasons for making this change are no longer a justification for it. However, it was convenient to maintain the new independent variable and the following first order, non-linear, differential equations were the ones which were actually integrated.

The three force equations become

$$\frac{d(V^2)}{ds} = - \frac{C_d(\alpha)S}{m}\rho(y)V^2 - 2(g_r\sin\gamma - g_\lambda\cos\gamma\cos\psi - g_\theta\cos\gamma\sin\psi)$$
$$- 2\omega^2 r\cos\lambda(\sin\gamma\cos\lambda + \cos\gamma\sin\lambda\cos\psi) \qquad (5.29)$$

$$\frac{d\gamma}{ds} = - \frac{C_L(\alpha)S\cos\varphi}{2m}\rho(y) - \left(\frac{1}{r} + \frac{g_r}{V^2}\right)\cos\gamma$$
$$- \frac{1}{V^2}(g_\lambda\sin\gamma\cos\psi + g_\theta\sin\gamma\sin\psi)$$
$$- \frac{\omega^2 r\cos\lambda}{V^2}(\cos\gamma\cos\lambda - \sin\gamma\sin\lambda\cos\psi)$$
$$- \frac{2\omega}{V}\cos\lambda\sin\psi \qquad (5.30)$$

$$\frac{d\psi}{ds} = \frac{C_L(\alpha)S\sin\varphi}{2m}\,\rho(y)\,\frac{1}{\cos\gamma} + \frac{\cos\gamma\sin\psi\sin\lambda}{r\cos\lambda}$$

$$- \frac{1}{V^2\cos\gamma}\,(g_\lambda\sin\psi - g_\theta\cos\psi) + \frac{\omega^2 r\cos\lambda}{V^2}\,\frac{\sin\lambda\sin\psi}{\cos\gamma}$$

$$+ \frac{2\omega}{V}\left(\frac{\sin\gamma\cos\psi\cos\lambda}{\cos\gamma} + \sin\lambda\right) \qquad (5.30)$$

The three positional equations become

$$\frac{dr}{ds} = -\sin\gamma \qquad (5.32)$$

$$\frac{d\lambda}{ds} = \frac{\cos\gamma\cos\psi}{r} \qquad (5.33)$$

$$\frac{d\theta}{ds} = \frac{\cos\gamma\sin\psi}{r\cos\lambda} \qquad (5.34)$$

Also, the differential equation for the time is

$$\frac{dt}{ds} = \frac{1}{V} \qquad (5.35)$$

5.5 Numerical Procedures

A major problem in the development of digital computer programs is the problem of retaining accuracy during the course of the computation while keeping the number of points (and therefore the computation time) as low as possible. In particular, the calculation of flight trajectories of reentry vehicles often requires an accuracy of six or eight figures in the altitude or geocentric radius, respectively. The latter requirement certainly exceeds the precision of numbers in most scientific computers when round-off error* inherent in all integration schemes is taken into account. Truncation-error is easily reduced by using one of the standard, high-accuracy integration techniques such as the fourth order Runge-Kutta method and the proper size of interval. However, round-off error proves much more difficult to eliminate, especially when a small enough interval size (and therefore a large number of points) is used to reduce the truncation-error. Consequently, a practical method must be found for reducing round-off error and also of reducing the number of integration points while simultaneously keeping the truncation-error to a minimum.

---

* Error is introduced from two sources:

(a) Truncation-error may be defined as the difference between the true value and the value obtained from the numerical integration formula which makes use of a finite number of terms from an infinite Maclaurin's or Taylor's series.

(b) Round-off error is the result of the limited accuracy (usually 8 figures) available within the computer.

The solution to the problem of round-off error took advantage of the basic characteristic of digital calculations. All numerical integration schemes involve adding small increments to the variables in order to find the next value after each step. Accordingly, these increments possess far more absolute accuracy than their respective variables. Thus, if the variables could take advantage of this increased absolute accuracy when incrementation took place, the round-off error would be greatly reduced. Notice that the incrementation stage would be the only one at which the increased accuracy (and therefore greater computation time) would be used since the calculation of the increments would use the variables only to their original accuracy.

In practical computational terms (using popular languages such as FORTRAN), the scheme merely involves the use of double-precision addition whenever the increments are added to the variables. Normal single-precision computation of the increments makes use only of the most significant part of the variables. In addition, an analysis of the whole program should be made as other sections may benefit from this technique. An example is the calculation of the altitude over an oblate Earth when only the geocentric latitude and the geocentric radius are available. Having calculated the "corrections" in single precision, they should be added in double precision.

A major reduction in the number of steps is also possible, by using a continuously varying interval step size dictated by truncation-error. It has been found that the truncation-error depends on how quickly the variables are changing as well as the step size. In the calculation of flight trajectories of entry vehicles using geographical coordinates, the key variables are the glide angle, the azimuth, and the latitude. Experience has shown that whenever all of these variables are reasonably stationary, the interval step size can be rather large. The criterion which was used to determine the interval size (within a certain preset range) was that the change in all these angles per step should be kept below a certain minimum value. By varying the interval size parameter, the value could be found for which the truncation-error was acceptable, since the double-precision incrementation always reduced the total round-off error to a negligible value.

Two basic methods were used to check the results of the computations. First, the atmosphere was removed (by setting the density to zero) and orbits were calculated for all the situations for which simple analytical solutions were known. Circular and elliptical orbits were compared over non-rotating and rotating Earth reference systems. In every case, the additional terms involving the Earth's rotation produced the proper result when changes were expected. On the other hand, parameters such as the orbital period and the major axis (for elliptical orbits) remained the same. Typical accuracies were six or seven figures over several orbits. These latter results also gave confidence in the numerical integration procedure referred to above. The check on the accuracy of Eq. (4.12) was made via the test program developed to check the original validity of its results.

In order to check the accuracy of the digital program when the atmosphere was present, the terms due to rotation and oblateness were removed. The equations were then further simplified (the atmosphere was made exponential, only zero roll angles were considered, and gravity was held constant at its average value) and the results compared with the output of the identical equations which had been wired into an analogue computer. Over the range for which the density approximation on the analogue computer was reasonably accurate (from 150,000 feet up to 300,000 feet) there was no discernable difference between the two results.

VI. RESULTS AND DISCUSSION FOR A CIRCULAR, NON-ROTATING EARTH

6.1 Introduction

The next three sections of this paper present an analysis of the performance of the CONTAC system with results obtained from the digital computer program listed in Appendix C. Section VI is devoted to presenting the overall capabilities of the CONTAC system as well as its limitations.

The method of presenting the results has been discussed in Sec. 1.3. Apart from the peak temperature, the two main quantities of interest are the equivalent down range and cross range. These values are illustrated and defined in Fig. 3.

In order to give the reader a better understanding of the way in which the CONTAC system works, a detailed history of all important variables is given in Figs. 11b and 11c. Figure 11a is a similar history of an uncontrolled reentry and may be used as a comparison. It is quite clear that the CONTAC system is easily able to control the temperature and prevent the additional rise of 650F° which occurred for the uncontrolled reentry.

In order to provide the details of the way in which the altitude angles are controlled, an expanded scale has been used for a portion of the reentry in Fig. 11d. Notice first that the line of time vs. down range is straight over this small portion of the reentry. Consequently, the abcissa of the graph has been labelled as time in addition to down range in order to provide a better feeling for the time scale of the cycling of the angle of attack by the CONTAC system. The most important information to be gained from Fig. 11d is the rates involved in the process. Notice that the fuel economy will be quite high due to infrequent use of the reaction jets.

6.2 Footprint Optimization

The standard reentry conditions at 350,000 feet altitude were chosen to be 26,000 feet per second for the speed and 1.0 degree for the glide angle. It might be noted that the speed is between 250 and 300 feet per second greater than the orbital velocity at this height, which therefore allows the spacecraft the possibility of having reached this position from an altitude of between 250 and 300 miles. A slight increase in the initial speed or glide angle would allow higher initial orbital altitudes up to 500 miles if that condition were desirable.

Two criteria were used in determining the best flight path which could be obtained from the initial conditions. The most important factor was to prevent throughout the reentry, the occurrence of a temperature greater than the peak value experienced during an equilibrium glide. In general, this requirement was slightly relaxed to allow temperature peaks which were greater by 20F° or 30F°. The second principal requirement was to obtain as large a cross range as possible throughout the whole extent of the down range capability.

In addition, some effort was made to increase the available down range increment. It is here that a further reason for the choice of 26,000 feet per second as the initial velocity was found. On Fig.9, it is shown that the speed lost up to the point of the first temperature maximum can easily be controlled to a value between 150 and 250 feet per second by the use

of the proper entry angle of attack. As noted in Sec. 1.9, this reduction in the vehicle's speed at the end of Phase II can be produced without causing a significant difference in altitude, glide angle, or down range distance at that position. This condition results from the variation of the drag coefficient with the angle of attack while the lift coefficient remains essentially constant. Consequently, due to the choice of speed at the initial position, the speed at the end of Phase II is either very slightly above (when low angles of attack with relatively low drag coefficients were used) or just below (when high angles of attack with much larger drag coefficients were used) circular orbital velocity. In a vacuum, the former speed results in a perigee condition which would produce an additional revolution around the Earth yielding 360 degrees of down range increment before the spacecraft descends to the same altitude. On the other hand, even in a vacuum, the latter speed would produce an apogee condition which brings about an immediate decrease in altitude. However, it is quite obvious that the presence of atmospheric drag will tend to alter the situation. Except when complicated by Earth oblateness, even the use of a very small angle of attack (at 5°, $C_D \approx C_{D_{min}}$) for the former situation results only in an extended glide at practically constant altitude (the rise is limited to 30,000 feet) until the speed falls below the circular value by about 1000 feet per second. On the other hand, the continued use of a high angle of attack for the latter situation reduces the speed very quickly resulting in very little incremental down range. In both extreme situations as well as any between, the incremental down range which is available by means of subsequent attitude control, is very nearly the same once the speed falls below the circular value by about 1500 feet per second. Therefore, any small speed variations at the end of Phase II on the order of one hundred or even fifty feet per second will result in large variations in incremental down range during Phase III of the reentry. As a consequence, the situation provides the guidance system with the capability of large down range increments, without interference from the CONTAC system, due to the possibility of an extended glide at very low angles of attack during which the speed is lost very slowly.

It should be noted that the above discussion is not altered by the inclusion of Earth rotation. The immunity is due to the fact that although the relative speed can change by up to $\pm$ 1500 feet per second at the equator for initially due West or due East reentries, respectively, the relative circular orbital speed also changes by exactly the same value. For other initial azimuths, the same relative change also occurs. Consequently, the same perigee and apogee conditions result upon the use of low and high entry (i.e., during Phase II) angles of attack, respectively.

With regard to satisfying the first criterion, the CONTAC system behaved very well. The operational limits were difficult to obtain since impractical situations had to be imagined before any degredation was found in the peak temperature performance. The discussion of the parameters which were varied during the reentry is confined to the attitude at this particular point; the result of the differences or errors in initial conditions is presented in Sec. 6.3.

The entry angle of attack was limited to a maximum of 65 degrees in order to prevent excessive heating rates at the rear of the spacecraft. It may be questionable whether larger pitch angles could be maintained due to problems of stability and control. However, assuming that these problems can be solved, there is no reason why angles as high as 75 or even 80 degrees cannot be used.

The lower limit for the entry angle of attack was approximately 35 degrees. When smaller values were used, the first maximum temperature was too large, since too little lift was developed to prevent the vehicle from descending too low on the initial penetration into the atmosphere.

The entry roll angle was always held at zero during this investigation. It is probable that a moderate roll angle during Phase II could increase the cross range without affecting the peak temperature control. Further investigation into this question might prove useful.

The skip angle of attack (which is controlled by the guidance system) is employed as the lower limit during Phase III. The largest value was the lesser of 55 degrees and the entry angle of attack. The lowest value used was 5 degrees. The guidance system might employ even smaller values down to zero if the upper surface of the spacecraft is not subject to or can cope with large heating rates. It should be noted that when the large entry angles of attack of 55 or 65 degrees were used, the CONTAC system often prevented the vehicle from assuming the skip angle of attack. As soon as it commences active operation at the beginning of Phase III, the CONTAC system disallows lower limits if they would permit too high a peak temperature. Again, it is noted that this operation is completely automatic.

The skip angle of roll (which is also controlled by the guidance system) is employed as the lower limit during Phase III. The "smallest" value was a full minus ninety degrees to the left (or plus ninety to the right). Smaller values were not used since they resulted in less cross range which was not in accordance with the second criterion. The "largest" value used was zero degrees. The CONTAC system often prevented the vehicle from assuming the skip bank angle when it was near ninety degrees; the conditions which led to this situation were the same as those which prevented very small skip angles of attack from being used and again the CONTAC system automatically disallowed the value and prevented the temperature from achieving too high a value.

The upper limit on the angle of attack during both Phases III and IV was 55 degrees. This value was fixed at the attitude which yielded the maximum vertical force. The upper limit for the bank angle was also the value which yielded maximum vertical force, namely zero degrees. The reasons for this choice have already been given in Sec. 1.10.

The lower limit for both the angle of attack and the bank angle did not affect the operation of the CONTAC system. Therefore, the choice of the values used will be discussed under the second criterion of cross range optimization.

While operating within the above values for the attitude, the CONTAC system performed the task of restraining the peak temperature experienced during the reentry without any degredation in performance for any combination of these values. As shown in Fig. 26, the peak temperature experienced during the reentry varied only $\pm$ 20 Fahrenheit degrees about the peak value experienced during an equilibrium glide. At the same time, cross range was practically constant (between 9 degrees and 11 degrees) throughout most of the available down range. Therefore, it is possible to obtain an "isothermal footprint" (with regard to the peak temperature experienced during the reentry) of considerable area when the CONTAC system is employed to constrain the temperature during

the flight. The dimensions are over one hundred degrees equivalent longitude in down range and about twenty degrees equivalent latitude in cross range. Considering the fact that the continental United States has a "length" of only forty degrees in equivalent longitude and is less than eighteen degrees of equivalent latitude wide, this "isothermal footprint" is about three times the area of that country. Consequently, during any given 12 (or at most 24) hour period, enough range control is available when using the CONTAC system to land at any point within or near the United States, as long as the terminal position is underneath or at least near the orbital envelope.

The attainment of the second criterion, the maximum possible cross range, was attempted by programming the skip bank angle (Phase III) and the lower limit for the angle of roll (Phase IV) within the restraints imposed by the first criterion. During each Phase of the reentry, several possibilites were investigated until the best overall performance was attained throughout the flight.

Note that the final results for cross range presented in this paper should not be considered the maximum possible values since it was not feasible to consider every parameter combination. In addition, the calculation was terminated at approximately 100,000 feet where the speed was about 3000 feet per second. As a consequence, about 3 degrees of equivalent latitude or about 200 miles of additional cross range is available before the final touchdown. However, as pointed out in Section 1.12, this final portion of the flight has not been included in the investigation. Therefore, no details for Phase V can be given.

On the other hand, density variations in the atmosphere and other sources of error may reduce the anticipated cross range. It was invariably found that when the relationships which yielded maximum cross range were incorrectly adjusted, the result was not only a decrease in cross range, but also either increased or decreased down range depending on the direction of the mismatch. Therefore, it may be necessary to sacrifice a portion of the cross range capability in order to retain a degree of down range mobility in the latter phases of the flight.

At this point, a detailed description of the methods used to maximize the cross range will be given for Phases I to IV.

During Phase I, no cross range control is available by means of attitude adjustments since atmospheric forces are negligible. In conjunction with the rotation of the Earth, it might seem possible to provide additional cross range by appropriately delaying or advancing the retro-fire time. For instance, in the case of a polar orbit with the landing zone near the equator, a retro-fire delay of 20 minutes would appear to provide 5 additional degrees of cross range. However, the results in Sec. 7.2 are contrary to such a possibility since the midline of the footprint closely follows the vacuum orbital track over the Earth's surface.

During Phase II, no cross range control was attempted. For entry angles of attack in the vicinity of 35 degrees, a very slight increase in cross range may be possible by trading off some vertical force for small amounts of side force. However, the speed is so great that only large forces will produce a turning radius with any appreciable effect. Thus, since large bank angles cannot be used without degrading the vehicle's peak temperature performance, there is too little to be gained from complicating the reentry with an additional parameter.

During Phase III, about ten to twenty percent of the cross range may be realized by using the optimum bank angle. Figure 27 shows that the best value for the skip angle of roll is about 75 degrees and this value was adopted for all subsequent trajectories. Another possibility for producing the most cross range was that of using a bank angle based on speed or perhaps altitude as suggested by Wagner (Ref. 6). However, the result was usually so close to 75 degrees that the additional complexity was not considered worth while.

It is during Phase IV that most of the cross range capability is realized. The lower limits for the angle of attack and the bank angle are not, of course, used until well after the beginning of Phase IV. This is because the CONTAC system retains control of the attitude until so much energy has been dissipated that the convective heating rate will continue to decline regardless of the attitude assumed. Once this condition has been reached, the guidance system will assume full control.

The basic concept behind the attempt to maximize the cross range is the relationship proposed by Wagner (Ref. 6). The problem in this investigation was how to implement the procedure on a practical basis. The method which was adopted is to base the maximum allowable bank angle (either left or right) on a variable connected with the shape of the trajectory. The three variables considered were the speed, the altitude, and the glide angle. The energy was also considered in order to combine both speed and altitude in some manner. However, the glide angle was rejected on the grounds that its variation during this part of the flight was an increase while a decrease in the bank angle was obviously required. This left the speed, altitude, and energy of the vehicle as a basis upon which to proceed.

The next step was to find the best angle of attack for the lower limit during Phase IV. The value found to yield the largest cross range was the angle for $(L/D)_{max}$ which occurs at 22.7 degrees for the vehicle under study. This value could be anticipated due to the fact that the maximum down range also results from the use of this angle.

At this point, a number of simultaneous tradeoffs were investigated. No attempt is made to justify the procedure nor even the final result. However, very briefly, it involved the determination of which power relationship (square root, linear, or quadratic) between the bank angle and one of the speed, the altitude, or the energy would yield the most cross range. The result that was consistantly best for all the reentries studied is the relationship shown in Fig. 28 in which the bank angle was a linear function of the speed over the interval shown. A few results were found for which a slightly greater cross range was achieved by using the square root of the speed over the specified interval; however, there was no way of predicting when the result would be larger. Therefore, the linear variation was adopted as the best solution.

In general, the above combination of the angle of attack for $(L/D)_{max}$ and the relationship shown in Fig. 28 for the bank angle as the lower limits during Phase IV proved a proper choice in one other way. In most situations, the azimuth change at 100,000 foot altitude was about 70 degrees. A slight additional increase until a speed of 1000 feet per second is reached during Phase V would yield a final azimuth change of about 80 to 90 degrees. It is certainly convenient to be heading in the proper direction to achieve the maximum possible additional cross range if it is required during the last stages of the reentry.

However, no calculations were made for Phase V of the reentry. But, it is worth mentioning that any procedures which are used to optimize the cross range must take into account the final approach and landing. Certainly, some cross range capability must be sacrificed in order to obtain a sensible landing pattern and a touchdown with a margin of safety and an allowance for adverse conditions at the landing site.

6.3 Effect of Non-Standard Initial Conditions

The results presented in this section deal with the effects produced by a difference between the desired and the actual initial values of the speed and the glide angle. The cause of the difference has been dealt with in some detail in Section 1.8. However, the investigation should not be confined only to that limited range of errors in attitude misalignment and incorrect retro-fire velocity increment that have occurred in recent flights. Therefore, the results of large deviations in the speed and the glide angle will be examined.

Before any results are presented, it should be noted that there are actually two problems to consider. The more important is whether the CONTAC system can cope with the deviations from the expected initial values. This question is easily answered. If the peak temperature during the reentry does not exceed by more than a small value the peak temperature which is experienced during an equilibrium glide, then this test is successful. If unsuccessful, the first temperature maximum is always higher than the standard peak temperature. On the other hand, when the test is successful, the peak which occurs during the flight (whether it be the first or a subsequent maximum) will be equal to the standard peak value.

The second question is whether or not the vehicle may still land at its original destination. Due to changes in the initial conditions, the shape, size, and position of the footprint is altered. The alterations are analyzed and the basic differences in the footprint are given. It can then be determined whether or not the footprint still contains the desired landing site.

Figure 29 shows the results of deviations in the initial speed which range from 25,700 to 26,300 feet per second. With respect to the first problem, it can be concluded that positive errors in the initial speed will not cause any degredation in the performance of the CONTAC system. However, peculiarly enough, it is the negative errors (due to a lower than normal initial speed) which cause some difficulty. For small errors, between zero and -100 feet per second, the increase in the peak temperature is also small, amounting to only $50F^{\circ}$ for most points on the footprint and only $90F^{\circ}$ for the extreme down range positions. However, for errors in the vicinity of -300 feet per second, the performance definitetly begins to decline; peak temperatures are $200F^{\circ}$ greater for most points on the footprint and at the extreme down range position, the peak is another sixty degrees higher still. Calculations could be made for even greater errors in the initial speed, but a simple check on the initial altitude from which such a reentry could be initiated reveals that even for an error of -300 feet per second (i.e. an initial speed at 350,000 feet of 25,700 feet per second), the maximum height from which the vehicle could have descended is only 130 miles. In addition, the errors in retro-fire alignment attitude would have to be about 30 degrees in alignment and about 250 feet per second in velocity increment. Such large errors would lie outside the normal

operating conditions. Finally, still larger errors (on the negative side at least) have no chance at all of occurring unless some deliberate action to decrease the speed is taken at an altitude below 100 miles. Since this height is below normal orbiting altitudes, the action would have to take place subsequent to retro-fire and is not considered a feasible possibility in the production of errors.

With regard to the position of the altered footprint, no problems are apparent. The maximum down range possible with the -300 feet per second error is still 60 degrees of equivalent longitude greater than the minimum down range with the +100 feet per second error. And with the latter error, global down range is already available. Therefore, with the +300 feet per second error, the guidance system will only have to take the vehicle around the Earth one additional orbit if the minimum down range position is past the landing site. With regard to cross range, it is quite clear that no problems exist here either. The minimum footprint width is still almost equal to the equivalent latitude span of the United States. Also, note that the minimum down range increment or footprint length is still greater than the equivalent longitude span of that country.

Figure 30 shows the results of deviations in the initial glide angle which range from 0.65 to 1.5 degrees. Reference 5 indicates that such an interval is far larger than would be expected due to normal errors in attitude alignment and velocity increment. However, Fig. 8 shows that this interval is to be expected if large retro-fire errors are present. Negative errors in the initial glide angle are easily handled by the CONTAC system and without any degradation in performance. But for positive errors even as small as 0.25 degrees, the CONTAC system fails to prevent the peak temperature from rising beyond the standard peak value. An analysis of the actual trajectories indicates that in every case the first temperature maximum is higher than the standard peak value due to the inability of the spacecraft to produce enough vertical force. As a result, the vehicle descends to a lower than standard height in the atmosphere at that position. The decrease in height results in a higher density without a compensating decrease in speed. Therefore, although the CONTAC system is not directly responsible for the temperature increase (recall that it does not go into active operation until the beginning of Phase III and these temperature peaks occur at the end of Phase II), nevertheless, the vehicle must be designed to withstand the additional rise in the peak temperature of 200F$^{\circ}$ if the initial glide angle may be as high as 1.25 degrees. However, it should be noted that the additional rise for an initial glide angle of 1.5 degrees is only about 300F$^{\circ}$ over the standard peak temperature. Consequently, the relationship is highly non-linear.

With regard to the position of the altered footprint there may be a problem for initial glide angles greater than about 1.4 degrees. Beyond this value, the maximum down range may be less than the minimum down range for initial glide angles less than 0.75 degrees. Whether the additional down range (obtainable from permitting the vehicle to travel once around the Earth in order to overlap the original down range position) can be usefully employed by the down range guidance scheme is a question which must be answered by simulated real time runs. The cross range is not a problem; there is more than the usual value.

A few additional points which apply to both Figs. 29 and 30 should be mentioned. First, there has been no attempt to optimize the upper and lower limits for the attitude when incorrect initial conditions in the speed or glide angle were present. Second, the results of an incorrect initial value for the down range position has not been taken into account. A one degree error in the initial down range would require about a twelve second error in retro-fire time assuming that the initial conditions of speed and glide angle were correct. However, the initial down range is also affected by the conditions which produce errors in the initial reentry values. For instance, an increased initial glide angle of 1.35 degrees (which results in a decreased down range over the whole footprint) is accompanied by a decreased down range at the initial position of eleven degrees of equivalent longitude when the retro-fire takes place at 250 miles. Consequently, it can be seen that the range control problem may be quite difficult to solve, but a solution should be available which is also compatible with the concept and operation of the CONTAC system.

VII. RESULTS AND DISCUSSION - OBLATE, ROTATING EARTH

7.1 Introduction

Section VI presented the problems associated with variations in the initial reentry conditions of speed and glide angle. However, it will certainly be possible to design for the spacecraft the equipment necessary to compute the proper attitude alignment and retro-fire velocity increment at the given orbital position. Therefore, these problems should not be considered to be very serious. On the other hand, the effects of Earth rotation and oblateness must be compensated for this task;may not oe simple in all situations.

In order to understand the individual effects of the Earth's rotation and oblateness, calculations were performed in which only one of these was present at a time. All the effects were then combined and the overall effect noted.

7.2 Effect of the Earth's Rotation

Figure 31 gives the results of calculations for a number of different reentry positions and directions when it was assumed that the Earth was circular, but rotating.

An analysis of the peak temperatures shows that the CONTAC system encounters a little difficulty in controlling the temperature when rotation and an initial Westerly heading along the equator combine to increase the initial relative speed between the vehicle and the atmosphere. On the other hand, this effect is reversed for an initial Easterly heading and lower than normal peak temperatures result . However, since the maximum difference in each case amounts to only $\pm$ 125 F^{o} for due West and due East equatorial reentries, respectively, the effect of rotation on temperature peaks cannot be considered a major problem. In most situations, the effect will probably be beneficial since the loss in inertial velocity during launch does not favour the establishment of orbits which travel in a Westerly direction against the rotation of the Earth.

With respect to the shape of the footprint, the most obvious part of the result is the shift towards negative values of cross range without any substantial change in down range. A comparison should be made(Fig. 29). In those cases, for a non-rotating Earth, errors in the initial speed of ± 300 feet per second produced large shifts in final down range. For these cases in which there is a ±1500 feet per second change in speed relative to the Earth due to an initial Westerly or Easterly heading, respectively, along the equator, the down range shift averages only about ± 8 degrees of equivalent longitude which is only about four percent of the total down range distance. It may be concluded, therefore, that for a given set of inertial initial conditions, the Earth's rotation is not a significant factor with respect to the final down range position of the vehicle. The physical explanation has already been presented in Sec. 6.2.

However, with regard to the cross range, an important shift occurs in the footprint. Fortunately, the extent of the shift is quite predictable. A comparison has been made with orbital tracks for vacuum trajectories. In almost every case studied, the extent of the shift is either the same as or algebraically less (i.e. whether the actual shift is positive or negative) by up to 1 degree than the value which would be found due to the rotation of the Earth under the equivalent circular orbit (Fig. 3). In most cases, the agreement was better than $\frac{1}{2}$ degree. The example easiest to visualize is the case for which the vehicle is initially heading due north from the equator. For a circular vacuum orbit through the initial position, the Earth will have rotated -10.8 degrees by the time the spacecraft reaches the equator again at 180 degrees down range. The actual shift in this case also amounts to -10.8 degrees. A check upon all other cases yields similar results.

The observed result may be explained as follows. In symmetric flight, with zero bank and sideslip, neither the lift nor the drag have a horizontal component perpendicular to V. Thus the only force available to produce horizontal curvature of the path (in an Earth-fixed reference frame) is the Coriolis force. This is the same whether the atmosphere is present or not, (except for the gradual reduction of V by the drag). Hence the horizontal curvature of the path is largely unaffected by the atmosphere.

7.3 Effect of the Earth's Oblate Shape

Figure 32 gives the results of calculations for a number of different reentry positions and directions when it was assumed that the Earth's poles were depressed to give the correct oblate shape, but that the gravitational field was purely a central force and the Earth was not rotating.

An analysis of the peak temperatures shows that the CONTAC system also encountered a little difficulty when the oblate shape of the Earth caused the effective initial glide angle (see Appendix B) to be larger than normal. For situations in which the effective initial glide angle is smaller than normal, the CONTAC system was able to maintain the peak temperature at the standard value.

The situation may be compared directly with the results of Fig. 30 in which both smaller and larger than normal values were used for the geocentric initial glide angle. The extreme negative difference between the effective and the geocentric initial glide angle is -0.18 degrees, and occurred for an initial latitude of 30 degrees North and an initial heading of due

North. The peak temperature for both this footprint and the one where the geocentric initial glide angle was 0.75 degrees is the normal peak value. In addition, the cross range is the same for each. However, down range achieved in the oblate Earth case is a relatively small but significant amount less at both ends of the footprint than for the case of the geocentric initial glide angle of 0.75 degrees. This difference is in good agreement with the expected result.

The extreme positive difference between the effective and the actual initial glide angle is +0.18 degrees, and occurred for an initial latitude of 60 degrees South and an initial heading of due North. For this reentry, the peak temperature for both this footprint and the one where the initial glide angle was 1.25 degrees is somewhat greater than the standard peak value, but the difference is, as expected, less for the case of the oblate Earth reentry. The peaks were about 160F° - 190F° greater. On the other hand, both the down range and the cross range achieved under both situations is substantially the same, which indicates that the continued decrease in the height due to the oblate shape has a continued effect on range while for peak temperatures, the effect ceases at the first temperature maximum (which occurs at about 18.5 degrees down range from the initial position).

The use of other geographical locations for the initial position produced similar results. A slight rise in the peak temperature was noted for the South-to-North trajectory, which produces an increase in the effective initial glide angle of only 0.07 degrees. The equator-to-pole reentry, which produces a decrease of 0.07 degrees, experiences a slight increase in down range.

In general, the results due to the shape oblateness of the Earth often showed that much of the effect can be accounted for by the difference between the effective and the geocentric initial glide angle. In addition, although there is a substantial difference in many situations between results for the same effective (over an oblate Earth) and geocentric (over a spherical Earth) initial glide angles, there is at least qualitative agreement in all cases. However, it should not be considered serious if the correction factors for non-normal values of the effective and the geocentric initial glide angles do not agree. The guidance system must reduce to zero either type of error. Since they are equal in magnitude, errors due to non-normal values in the actual initial glide angle will provide the necessary information on the extent of the errors which may be expected on account of the shape oblateness of the Earth.

There is another possible approach to the problem of shape oblateness. Rather than relying on the CONTAC - guidance system to correct for the errors introduced by incorrect values in the effective initial glide angle, it should be possible to produce the correct down range by an appropriate adjustment of the actual geocentric initial glide angle. The adjustment would have to take account of the geographical location of the initial position. The change in the actual initial glide angle could be produced by correcting the retro-fire attitude alignment and velocity increment.

7.4 Effect of the Earth's Oblate Gravity Field

The results of including the oblateness term in the gravitational field for the Earth showed a negligible difference between the various foot-

prints which result from different initial positions and initial heading directions. Consequently, it may be concluded that even the $C_{2,0}$ coefficient has a negligible effect on the reentry of a spacecraft and that all other terms may be completely ignored.

One minor effect was found which tends to cancel the shape oblateness contraction in down range which resulted when proceeding from polar to equatorial regions. For these cases, the small reduction in gravitational force on the vehicle due to the additional distance between the vehicle and the equatorial bulge resulted in a small increase in down range of the order of 5 degrees at the minimum down range position and 10 degrees at the maximum down range position. Since this increase amounted to less than 5% of the total, and no other results were affected, it was not considered significant. Note that in the case of shape oblateness, the decrease in down range was double this value when the initial position was at the pole.

7.5 Effect of the Earth's Rotation and Oblateness

Figure 33 gives the results of calculations for a number of different reentry positions and directions for a rotating oblate Earth.

An analysis of the peak temperatures shows that the CONTAC system again encountered the same expected difficulty in controlling the peak temperature when rotation and the initial heading combine to increase the relative speed between the vehicle and the atmosphere. In these situations, an initial heading which is due West together with an initial equatorial position results in an increase in the peak temperature of about 100F° above the standard peak temperature. Almost exactly the same result occurred in the case of a circular, but rotating Earth. Also, the decrease of 100F° due to an initial Easterly heading is again present.

In addition, the expected increase in the peak temperature also occurs when the effective initial glide angle is larger than normal although in the rotating Earth environment the increase is relatively small. The worst situation studied occurred for an initial latitude of 60 degrees South with an initial heading of due North; for this case, the rise was only a modest 100F° over all but the extreme down range portion of the footprint for which a rise of 200F° was incurred.

It is conceivable that a higher temperature increase would result if the initial heading and initial latitude combined to produce both a higher than normal effective initial glide angle together with a higher than normal effective initial speed. However, in order to produce the former, almost due North initial headings are required while the latter requires almost due West values. The latitude must also be large if the former is to be significant, which in turn reduces the latter. Therefore, it is deemed quite unlikely that peak temperature will be more than 100F° above the standard peak during reentries for which the inertial initial conditions are a speed of 26,000 feet per second and a flight path angle of 1.0 degrees. Finally, even if the effective initial glide angle is greater than 1.0 degrees and increases in the peak temperature higher than 100F° occur, it would be possible to compensate for this in advance of the reentry by the appropriate adjustment in the retro-fire attitude alignment and velocity increment.

With regard to the shape and position of the footprint, the combined effects of rotation and oblateness upon both the down range and the cross range do produce a situation which is difficult to analyze. However, the results may be considerably simplified with the aid of Figs. 31 and 32 and the analysis which applies to each. Very broadly speaking, the two effects seem to combine in an almost linear fashion and produce the range distortions associated with shape oblateness together with the cross range shift due to rotation. For example, the rotational case used in Sec. 7.2 in which the vehicle is initially heading due North from the equator results in the identical cross range shift of 10.8 degrees of equivalent latitude at 180 degrees of equivalent longitude down range; in addition, this same case now manages to have an extended footprint with almost the same maximum down range as the oblate shape case. Continuing with this same footprint, the cross range shift at 360 degrees of equivalent longitude down range is 21.4 degrees as compared to 21.7 degrees for the circular orbit in vacuum which goes through the initial position. The above results are illustrated in Fig. 33a. Many more examples (there were 40 pairs of point to choose from as well as the general outline of each footprint) far too numerous to mention bear out the conclusions that the effects of rotation and oblateness combine with very little interaction.

In conclusion, it should be noted that the effects of both rotation and shape oblateness upon the shape and size of the footprint are large. The effect of gravitational oblateness is usually negligible and is so completely masked by shape oblateness that it is never significant. With regard to shape oblateness, the primary effect is to change the down range position and span of the footprint without producing any major change in the cross range. On the other hand, rotation produces a dramatic cross range shift without any major effect upon the down range. However, the cross range shift due to rotation is accurately predicted by the vacuum orbital track over the Earth's surface of an object which passes through the initial position of the vehicle. It must have the same azimuth and also possess the correct speed (the glide angle should be zero) relative to the Earth's surface to place the object in a circular orbit. Consequently, although the cross range shift is quite large and cannot be ignored if accurate navigation is to be achieved, the guidance system will be able to predict the extent of the shift as a function of the down range.

VIII. RESULTS AND DISCUSSION - DENSITY PERTURBATIONS

8.1 Introduction

As noted in Sec. 4.6, there is very little data available on spatial density variations in the Earth's atmosphere. As a result, several different types of variations were proposed and calculations carried out to determine if they caused any degradation in the CONTAC system performance. Each sub-section gives a detailed account of the extent of the variation for which the CONTAC system could correct, so that its limitations are established.

8.2 Effect of Different Basic Models

An attempt was made to determine just how small the atmospheric scale height had to be before the CONTAC system would be unable to correct for larger than normal density gradients. The details of the density versus altitude relationship are shown in Fig. 34. Scale heights between 14,000 feet and 26,000 feet were used during the initial portion of the reentry. It can be

seen that the increase in density at 230,000 feet due to the use of the smallest scale height produced a density increase of about 250% which is an extreme value in view of the available data (Ref. 33).

The results were very satisfactory. The use of the CONTAC system resulted in no increase in the peak temperature larger than 10F° above the standard peak value and the range results were within 2% of the values found when using the U.S. Standard Atmosphere, 1962. The 1959 ARDC Standard Atmosphere was also used. Again, as expected, there was no effect noted.

8.3 Effect of Altitude-Dependent Perturbations in the Density

In order to simulate variations dependent upon the altitude, information found in Ref. 32 was adopted. The resemblence to a sine wave suggested the curve shown in Fig. 35. The variations are all about the U.S. Standard Atmosphere, 1962. After the reentry was calculated for this shape, a number of others were investigated. The wave length was varied from 25,000 feet to 100,000 feet. The amplitude was increased to give a fractional variation of ± 0.2 and the nodal point was shifted to test the sensitivity to all phase relationships.

The most severe case used a wave length of 25,000 feet and an amplitude of 0.2 for the fractional variation. The above values together with a scale height of 18,000 feet for parts of the U.S. Standard Atmosphere, 1962 yield a density increase of 45% in an altitude range of 4,250 feet in the region one-half of a wave length from the nodal point. Since Ref. 32 indicates that the"density may vary by as much as 50 percent when descending through an altitude increment of 4,250 feet" and that this is the maximum increase to be expected, it was felt that further calculations were not necessary.

Again, the results were completely satisfactory in that the CONTAC system was able to maintain control of the temperature. The down range, however, did vary by as much as 5% from the values obtained by using the U.S. Standard Atmosphere, 1962.

8.4 Effect of Random Density Perturbations

The simulation of random spatial density variations presented an unavoidable difficulty. Data is not available from which to set up a realistic model of the variations which might be expected. A little information combined with intuitive judgements was used to postulate a random variability model based upon the distance travelled by the spacecraft during the reentry. The detailed description of the model is given in Sec. 2.6.

The results may best be described by reference to Fig. 36. The combination of variance and correlation interval above the hatched zone usually resulted in situations with which the CONTAC system could not cope. Increases in the peak temperature of about 300F° usually resulted for variances which were 0.05 to 0.1 greater than the variance shown acceptable for a given correlation interval. For combinations in the hatched zone, the peak temperature was usually less than 25F° greater than the standard peak value; however, in isolated examples, an increase of between 50F° and 100F° was experienced. For variances which were 0.05 to 0.1 less than the variance immediately under the hatched region, the peak temperature was always equal to the standard peak value.

An exhaustive study could not be made due to the statistical nature of the problem. A complete set of results would have required over a thousand individual runs, each requiring up to three or four minutes of computer time.

A large number of runs is needed because a probability analysis is required. For each combination of variance and correlation interval, a number of different random density distributions would have to be used, and a graph deduced of the probability of exceeding a given temperature for that combination. Repeating for a number of variance-correlation pairs would provide the data for a Figure like 36, in which the hatched area would be bounded by curves of constant probability of exceeding a set temperature, e.g., 1% and 10% probability of exceeding a 300F° rise. In order to have a reasonable degree of confidence in the results, a great many individual runs must be made for each pair of values for the variance and correlation interval. Since certain results would occur infrequently, the suggested figure for the number of runs in each set is one hundred. It seems likely that al least ten pairs or sets would be required. For the graph presented, about $2\frac{1}{2}$ hours of time was used to make about 50 individual runs for 7 pairs of values for the variance and correlation interval.

One additional factor of importance must be noted. In addition to the radiation-equilibrium temperature, the actual temperature based on Eq. (3.6) was calculated for the nose stagnation point. In general, the equilibrium temperature is held at a constant value by the CONTAC system for a lengthy period. For this situation, the difference between the real peak temperature (in which the skin possesses some heat capacity) and the equilibrium peak temperature is usually less than 5F°. For this reason, the real peak temperature has heretofore been ignored. However, when random spatial density variations were encountered which had a correlation interval less than 200 miles (and especially when this distance was less than 100 miles - at 20,000 feet/sec, 100 miles are traversed in only 26 seconds), there was often a large difference between the two peak values. In one instance when the equilibrium peak temperature was about 300F° greater than the standard value, a heat capacity of 2.0 BTU/ft^2 (which might be provided by a 0.03 inch thick skin of Molybedenum weighing 1.6 lb/ft^2) reduced the excess temperature by 20% to about 240F°. Only a few of the 50 runs provided a situation for which this result could occur. Consequently, no definitive conclusions could be drawn. But it will certainly be very fruitful to determine the extent to which even a limited heat capacity for the vehicle's skin can reduce excess peak temperatures caused by random spatial density variations in the Earth's atmosphere.

IX. CONCLUSIONS

9.1 Validity of the CONTAC System Concept

The results of this investigation indicate that the CONTAC system which has been proposed is a highly effective method for reducing the peak temperature experienced during the reentry of a manned, lifting spacecraft. The maximum temperature which a vehicle experiences during a fixed atttitude reentry is lowest for an equilibrium glide. This maximum is taken as the standard peak value. The CONTAC system prevents the actual peak temperature experienced during the reentry from exceeding that value as long as the initial conditions are not such as to produce a value for the first temperature maximum

that is higher than the peak value for the equilibrium glide. In addition, if the first peak temperature is higher, then the CONTAC system is able to prevent all subsequent peaks from exceeding the first.

With regard to range control, the CONTAC system is intended for use with any type of guidance scheme which uses closed loop control. The available down range increment for the vehicle in this study (which is similar to the M2-F2 now currently under investigation) is in excess of 100 degrees of equivalent longitude. The combined cross range for turns both to the left and to the right is greater than 20 degrees of equivalent latitude throughout all of the above value for down range. Within these limits, the footprint is essentially isothermal, i.e. the temperature experienced during the reentry is the same regardless of the terminal position of the vehicle within the footprint.

9.2 Effects of Adverse Conditions on the Reentry

The performance of the CONTAC system has been investigated under several types of adverse conditions. The first type is errors in the initial values of speed and glide angle. The second consists of those effects which can be predicted in advance, such as the Earth's rotation and oblateness. The third type consists of variations from standard values of the atmospheric density which are both altitude dependent and random.

Large errors in the initial speed and glide angle are considered unlikely since they can be produced only by extremely large retro-fire errors. (Past experience shows that retro-fire errors can be kept quite small.) At the deboost position, attitude alignment must be in error by about 5 degrees or the velocity increment must be in error by about 50 fps before any significant changes in the initial speed and glide angle are noted at the reentry altitude of 350,000 feet. However, if large retro-fire errors are present, the following results are noted. Negative errors in the initial speed up to 100 feet per second (larger errors are improbable) could cause an increase in the peak temperature up to 100 F^{o}. However, an increase in the initial glide angle of 0.5 degrees causes an increase of 300 F^{o} in the peak temperature.

The Earth's rotation causes a maximum increase of about 100 F^{o} in the peak temperature when the Earth's rotation and the initial heading combine to produce an effective initial speed 1500 feet per second above the standard value of 26,000 feet per second. When the Earth's oblateness and the initial heading and latitude combine to produce an effective initial glide angle of 0.18 degrees greater than the standard value of 1.0 degree, the maximum increase in the peak temperature is again about 100 F^{o}. However, the rotational condition is unlikely to occur in practice since satellite orbits are usually in the same sense as the Earth's rotation, and the oblateness effect may be compensated for in advance of the reentry by an appropriate adjustment of the retro-fire attitude alignment and velocity increment.

Atmospheric density perturbations based on altitude did not, within reasonable limits, produce any situations for which the CONTAC system could not correct. The maximum likely vertical density gradients had no effect on the CONTAC system performance. On the other hand, random spatial (roughly horizontal) density variations produced a very different result. The investigation described a basis for judging the adequacy of the CONTAC system when sufficient information about random density variations in the Earth's atmosphere become available. Insufficient data could be found for the altitudes of concern, namely between 220,000 feet to 290,000 feet, in order to arrive at a firm conclusion at this time.

REFERENCES

1. Wingrove, R.C. — A Survey of Atmospheric Reentry Guidance and Control Methods, IAS 31st Annual Meeting, January 21-23, 1963, New York, N.Y., IAS Paper No. 63-86.

2. Bryson, A.E., Denham, W.F., Carrol, F.J., Mikami, Kinya — Determination of Lift or Drag Programs to Minimize Re-Entry Heating, Journal of the Aerospace Sciences, Vol. 29, No. 4, April, 1962.

3. Levinsky, E. S. — Application of Inequality Constraints to Variational Problems of Lifting Re-Entry, IAS 29th Annual Meeting, January 23 - 25, 1961, New York, New York, IAS Paper No. 61-21.

4. Stalony-Dobrzanski, J. — Re-Entry Guidance and Control Using Temperature Rate Flight Control System, AIAA 2nd Aerospace Sciences Meeting, January 25-27, 1965, New York, N.Y., AIAA Paper No. 65-47.

5. Collins, R. R., Dotts, H.W., Hoyler, W.F., Hecht, K.F. — Guidance, Control, and Propulsion Systems, Gemini Mid-program Conference Including Experimental Results, February 23-25, 1966, Manned Spacecraft Center, Houston, Texas, NASA SP-121.

6. Wagner, W.E. — Roll Modulation for Maximum Re-Entry Lateral Range, Journal of Spacecraft, Vol. 2, No. 5, September-October 1965.

7. Smith, R.H., Menard, J.A. — Supercircular Entry and Recovery with Maneuverable Manned Vehicles, National IAS, ARS Joint Meeting, June 13-16, 1961, Los Angeles, California, IAS Paper No.61-114-1808.

8. Gaines, L.M., Surber, T.E. — Prediction of Optimum Approach and Landing Techniques for Manned Re-Entry Gliders, National IAS, ARS Joint Meeting, June 13-16, 1961, Los. Angeles, Calif. IAS Paper, No. 61-114-1809.

9. Thompson, M.O., Weil, J., Holleman, E.C. — An Assessment of Lifting Reentry Flight Control Requirements during Abort, Terminal Glide, and Approach and Landing Situations, AGARD Specialists' Meeting on Stability and Control, Sept. 20-23, 1966, Cambridge, England.

10. Etkin, B. — Longitudinal Dynamics of a Lifting Vehicle in a Circular Orbit, UTIAS Report No. 65, AFOSR TN 60-191, Feb. 1960.

11. Drummond, A.M. — Performance and Stability of Hypervelocity Aircraft on a Minor Circle, UTIAS Report to be published.

12. Technology Week — Including Missiles and Rockets, American Aviation Publications, Inc., Vol. 19, No. 20, July 25, 1966.

13. Geiger, R.E. — Experimental Lift and Drag of a Series of Glide Configurations at Mach Numbers 12.4 and 17.5, IAS National Summer Meeting, June 28-July 1, 1960, Los Angeles, California, IAS Paper No. 60-92.

14. Felterman, D.E. Neal, L, Jr. — An Analysis of the Delta-Wing Hypersonic Stability and Control Behaviour at Angles of Attack Between 30^{o} and 90^{o}, NASA TN D-1602, March 1963.

15. Wells, W.R. Armstrong, W.O. — Tables of Aerodynamic Coefficients Obtained from Developed Newtonian Expressions for Complete and Partial Conic and Spheric Bodies at Combined Angles of Attack and Sideslip with Some Comparisons with Hypersonic Experimental Data, NASA TR R-127, 1962.

16. Smith, R.H. Menard, J.A. — Supercircular Entry and Recovery with Maneuverable Manned Space Vehicles, IAS-ARS Joint Meeting, June 13-16, 1961, Los Angeles, Calif., IAS Paper No. 61-114-1808.

17. Chapman, D.R. — An Analysis of the Corridor and Guidance Requirements for Supercircular Entry into Planetary Atmospheres, NASA TR R-55, 1960.

18. Mugler, J.P. Jr. Olstad, W.B. — Static Longitudinal Aerodynamic Characteristics at Transonic Speeds of a Blunted Right Triangular Pyramidal Lifting Reentry Configuration for Angles of Attack up to 110^{o}, NASA TN D-797, June 1961.

19. Fournier, P.G. — Wind-Tunnel Investigation at High Subsonic Speed of the Static Longitudinal Stability Characteristics of a Winged Reentry Vehicle Having a Large Negatively Deflected Flap-Type Control Surface, NASA TN D-2030, October, 1963.

20. Ware, G. M. — Low-Subsonic-Speed Static Stability of Right-Triangular-Pyramid and Half-Cone Lifting Reentry Configurations, NASA TN D-646, February 1961.

21. Crabtree, L.F. Dommett, R.L. Woodley, J.G. — Estimation of Heat Transfer to Flat Plates, Cones and Blunt Bodies, Royal Aircraft Establishment Technical Report No. 65137, July 1965.

22. Leo Rute — A Study of Aerodynamic Effects of Isothermal and Temperature Gradient Atmospheres on Reentry Trajectories, Polytechnic Institute of Brooklyn Department of Aerospace Engineering and Applied Mechanics, PIBAL Report No. 738, March 1962.

23. Parvin, R.H. — The Earth and Inertial Space, Part I - Motions of the Earth, Aerospace Engineering, Vol. 18, No. 4, April 1959.

24. Kozai, Yoshihide — Numerical Results on the Gravitational Potential of the Earth from Orbits, Proceedings of the First International Symposium on the Use of Artificial Satellites for Geodesy, April 26 - 28, 1962, Washington, D.C.

25. Kaula, W.M. — Tesseral Harmonics of the Gravitational Field and Geodetic Datum Shifts Derived from Camera Observations of Satellites, NASA TN D-1848, June 1963.

26. Kaula, W.M. — Improved Geodetic Results from Camera Observations of Satellites, Journal of Geophysical Research, Vol. 68, No. 18, Sept. 15, 1963.

27. Wagner, C.A. — The Drift of a 24-Hour Equatorial Satellite Due to an Earth Gravity Field Through 4th Order, NASA TN D-2103, February 1964.

28. Wagner, C.A. — The Equatorial Ellipticity of the Earth as Seen from Four Months of Syncom II Drift Over the Western Pacific, NASA TN D-3313, July 1966.

29. Wagner, C.A. — The Equatorial Ellipticity of the Earth from Two Months of Syncom II Drift Over the Central Pacific, NASA TN D-3315, July 1966.

30. Baker, R.M. Jr. — Three-Dimensional Drag Perturbation Technique, ARS Journal, Vol. 30, No. 8, August 1960, AFOSR TN 59-767, July 1, 1959.

31. Staff — U.S. Standard Atmosphere, 1962.

32. Cole, A. E.
Kantor, A.J. — Horizontal and Vertical Distribution of Atmospheric Density Up to 90 km, Air Force Surveys in Geophysics, No. 157, AFRCRL-64-483, June 1964.

33. Cole, A.E.
Court, A.
Kantor, A.J. — Model Atmospheres (Chapter 2, Handbook of Geophysics and Space Environment), Air Force Cambridge Research Laboratories, L.G. Hancom Field, Bedford, Massachusetts.

34. Etkin, B. — Private Communications, 1966.

35. Sommer, S.C.
Short, B.J. — Point Return from a Lunar Mission for a Vehicle that Maneuvers Within the Earth's Atmosphere, NASA TN D-1142, November 1961.

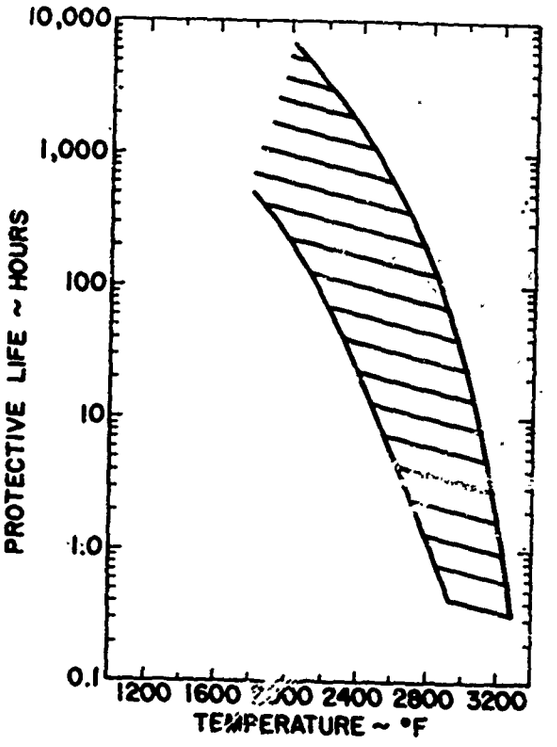
36. Nielsen, J.N.
Goodwin, F.K.
Mersam, W.A. — Three-Dimensional Orbits of Earth Satellites, Including Effects of Earth Oblateness and Atmospheric Rotation, NASA MEMO 12-4-58A, December 1958.

37. Fine, J.H. — Stability of Flight Paths of Lifting Vehicles During Entry into Planetary Atmospheres, UTIAS Technical Note No. 48, July 1961, AFOSR 1488.

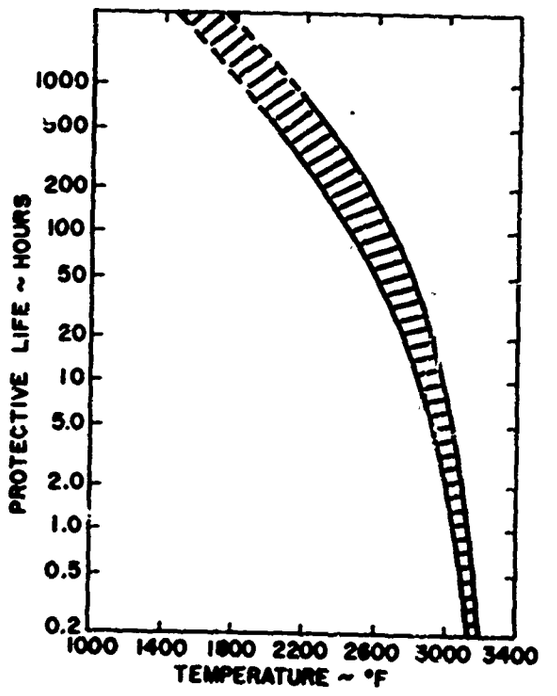
38. Brown, R.C.
Brulle, R.V.
Griffin, G.D.
Six-Degree-of-Freedom Flight Path Study Generalized Computer Program, Part I, Problem Formulation, WADD Technical Report No. 60-781, Part I, May 1961.

39. Etkin, B.
Dynamics of Flight, J. Wiley & Sons, Inc., New York, 1959.

40. Etkin, B.
Dynamics of Flight, Appendix A-6, - Derivative of a Vector.



Protective capability of silicide-base coatings for molybdenum.



Protective capability of the chromium-thorium-silicon coating for columbium alloys.

FIGURE 1 ENDURANCE TIME vs TEMPERATURE FOR COATED REFRACTORY METALS

(From 'Refractory Metals and their Protection', Thomas D. Cooper and Oscar O. Sharp, Aerospace Engineering, Vol. 22, No. 1, January 1963.)

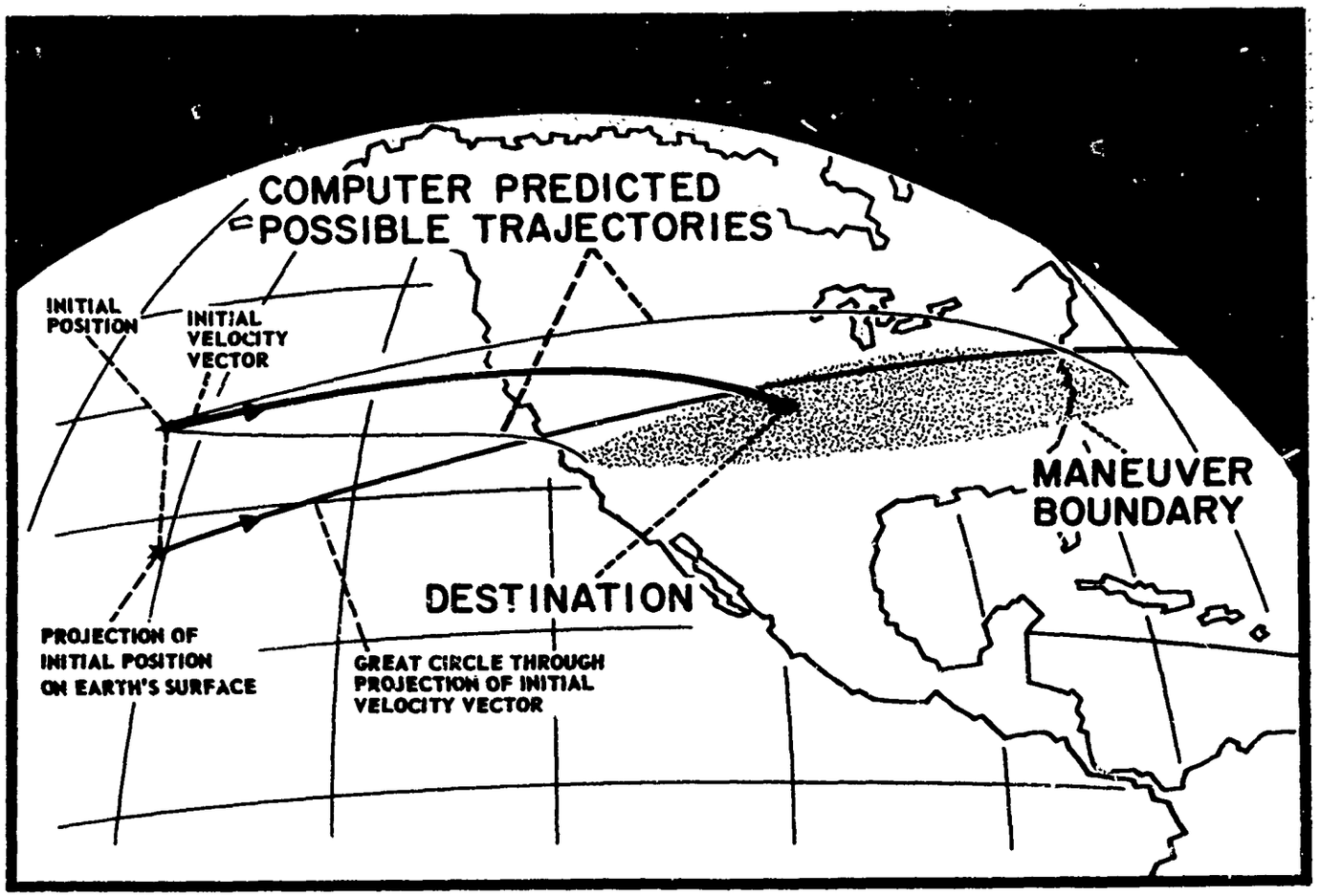
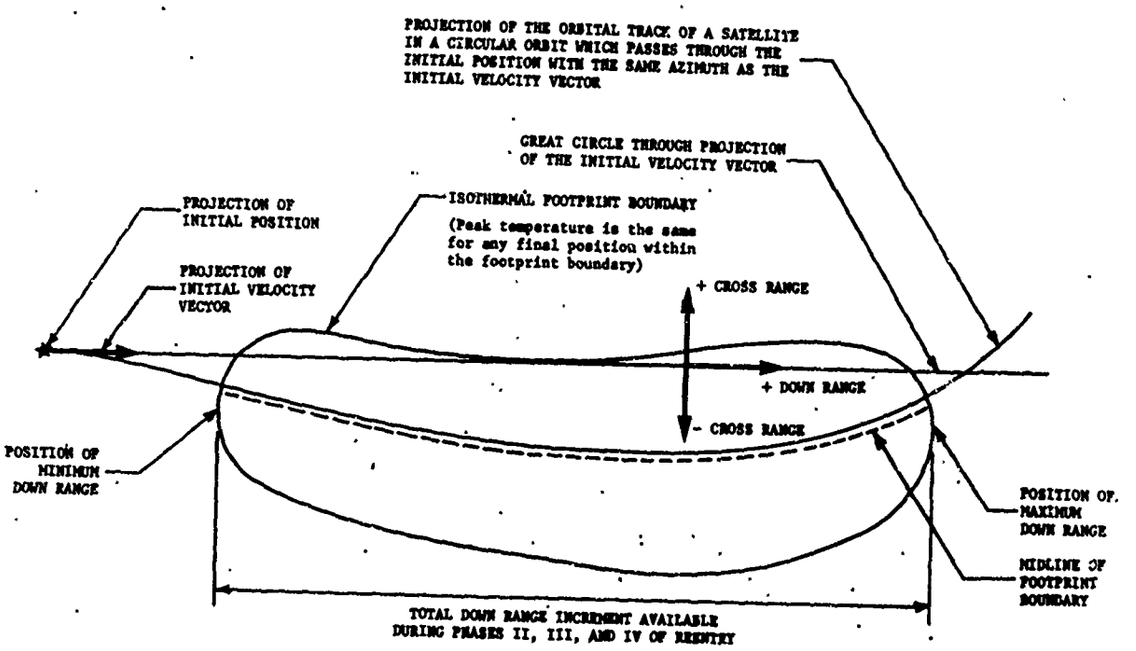


FIGURE 2 TYPICAL MANOEUVRE BOUNDARY OR FOOTPRINT

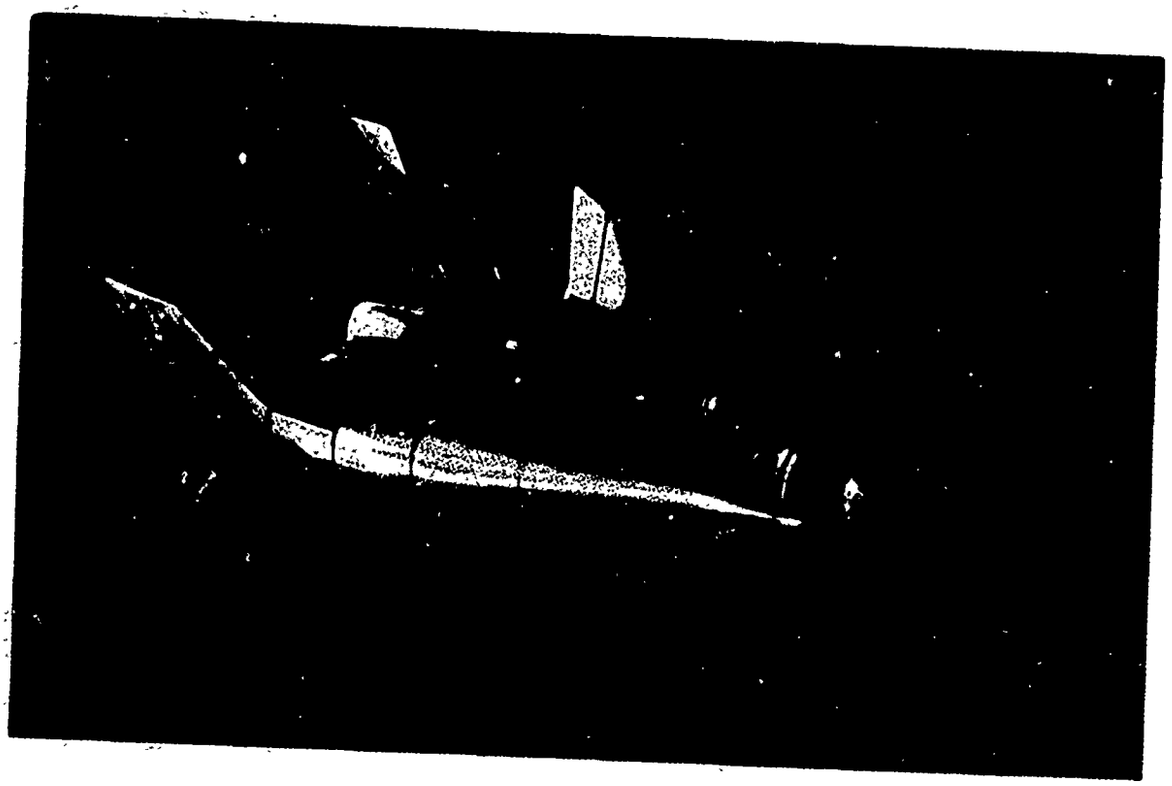
(From 'A Survey of Atmospheric Reentry Guidance and Control Methods', Rodney C. Wingrove, IAS 31st Annual Meeting, January 21–23, 1963, New York, N.Y., IAS Paper No. 63–86.



NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Down range is always positive. It is in the direction parallel to the great circle.
3. Cross range is measured in the direction perpendicular to the great circle.
4. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)

FIGURE 3 FOOTPRINT TERMINOLOGY



HL-10 Artistic Representation
Northrop Norair, Hawthorne, California
January, 1966

FIGURE 4 TYPICAL VEHICLE CONFIGURATION

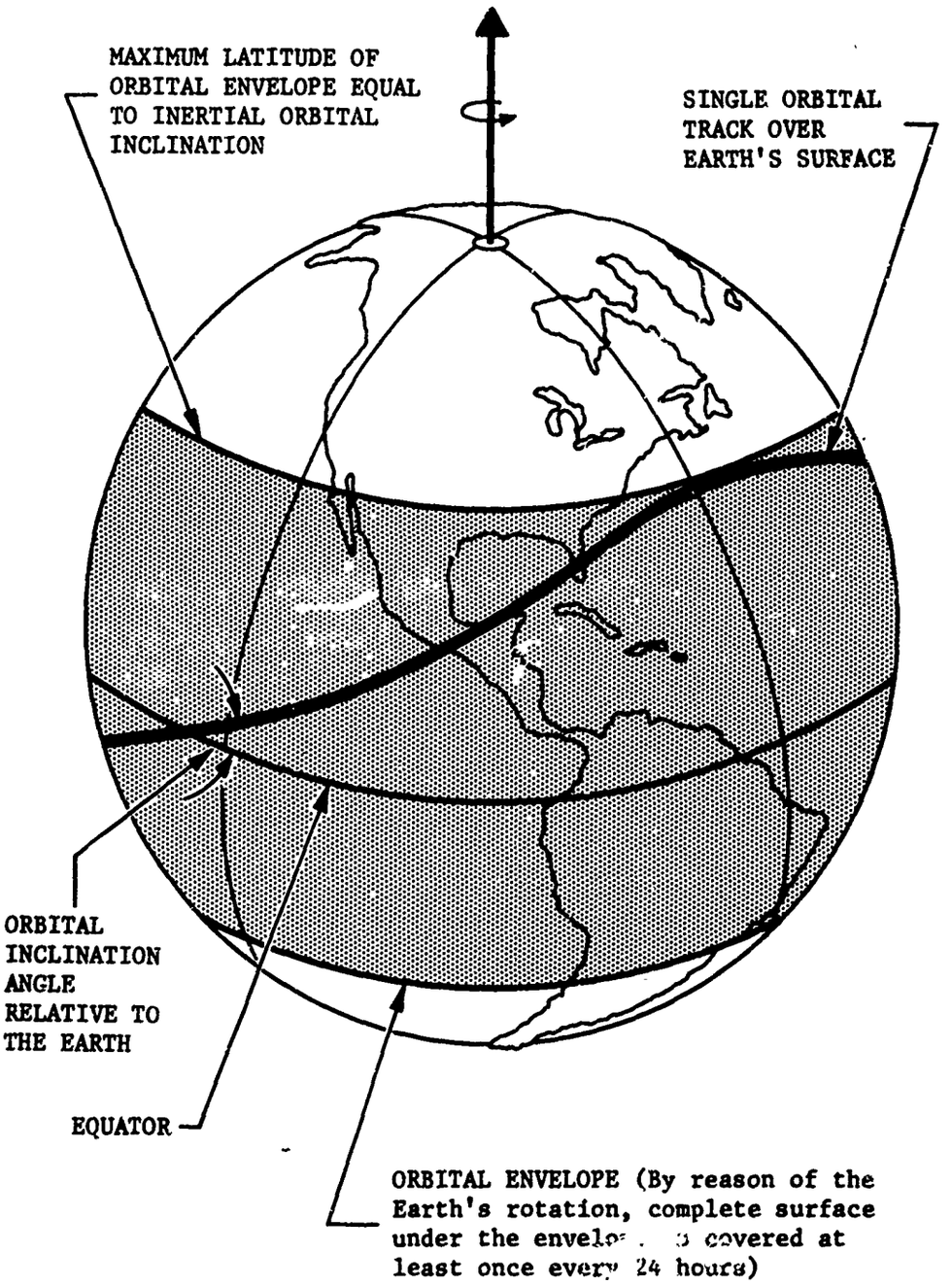


FIGURE 5 ORBITAL ENVELOPE

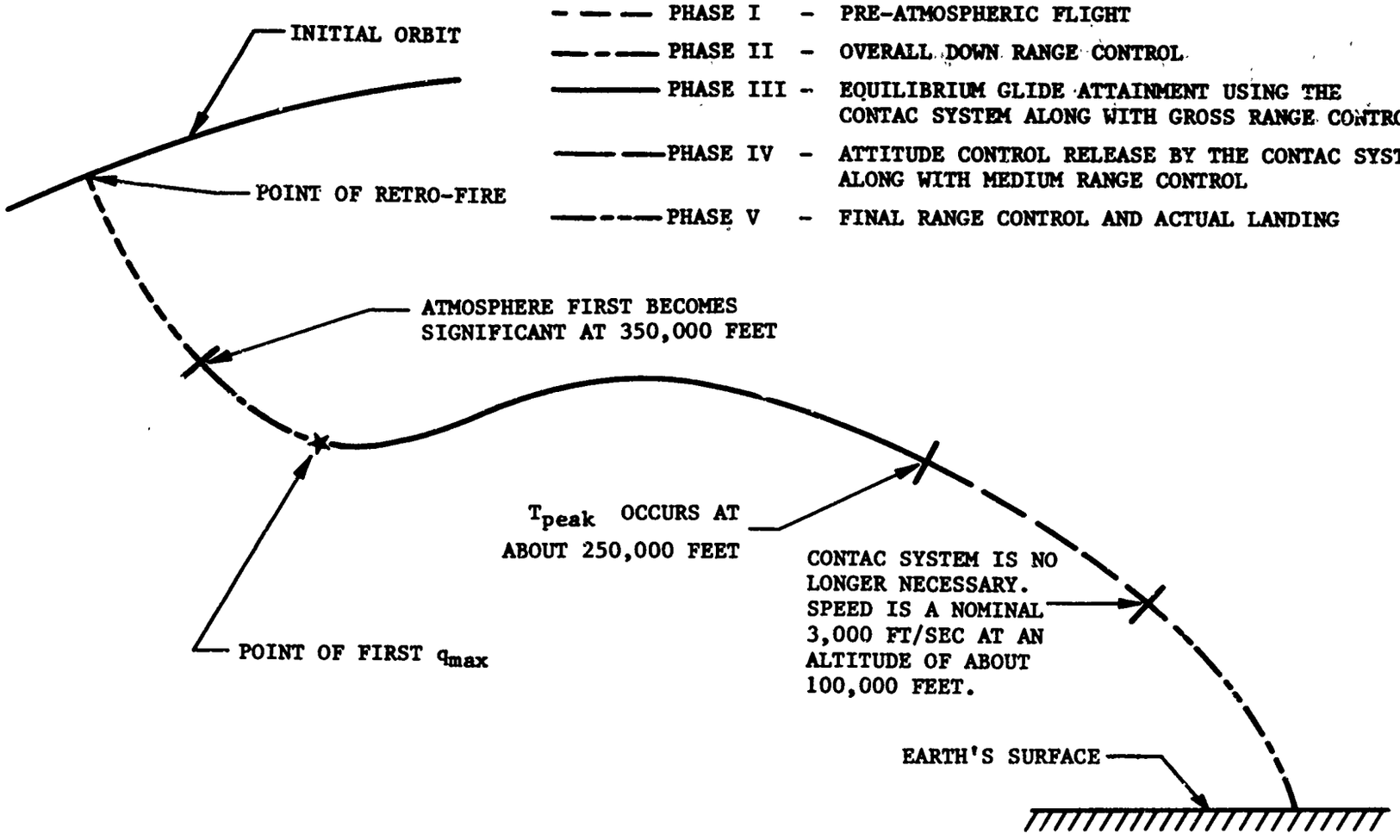


FIGURE 6 THE FIVE REENTRY PHASES
(Not To Scale)

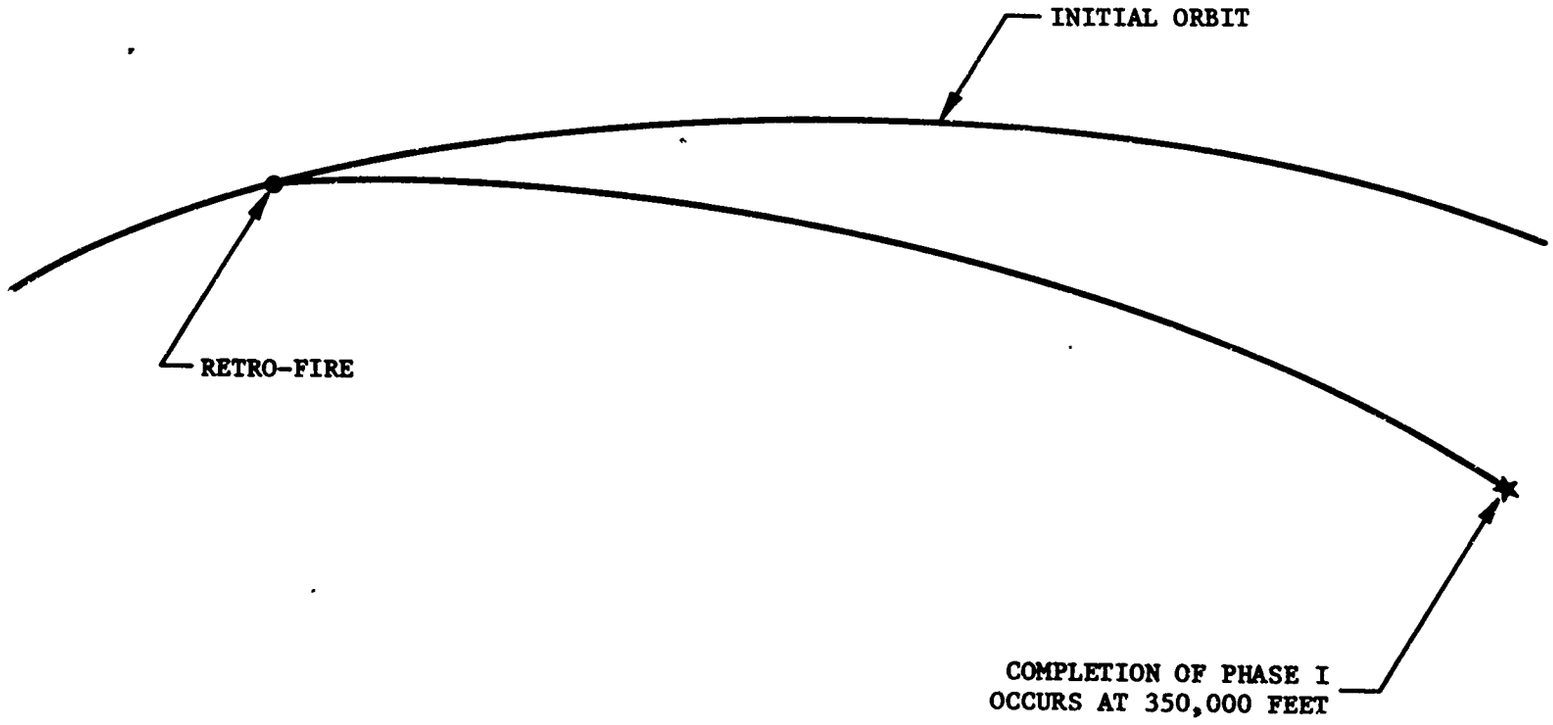


FIGURE 7 PHASE I - PRE-ATMOSPHERIC FLIGHT

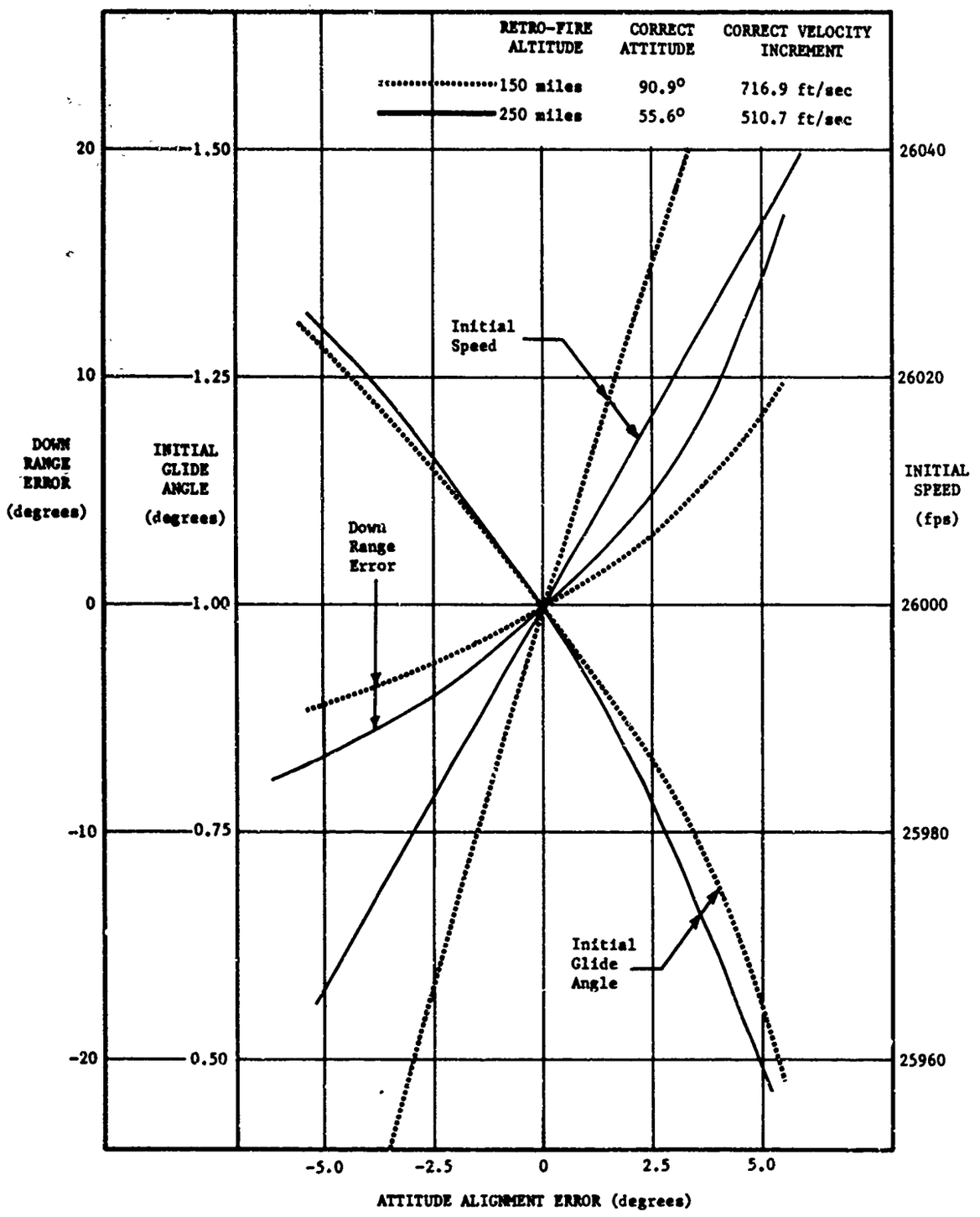


FIGURE 8a INITIAL CONDITIONS AT 350,000 FEET vs RETRO-FIRE ATTITUDE ALIGNMENT ERROR FOR AN INITIALLY CIRCULAR ORBIT

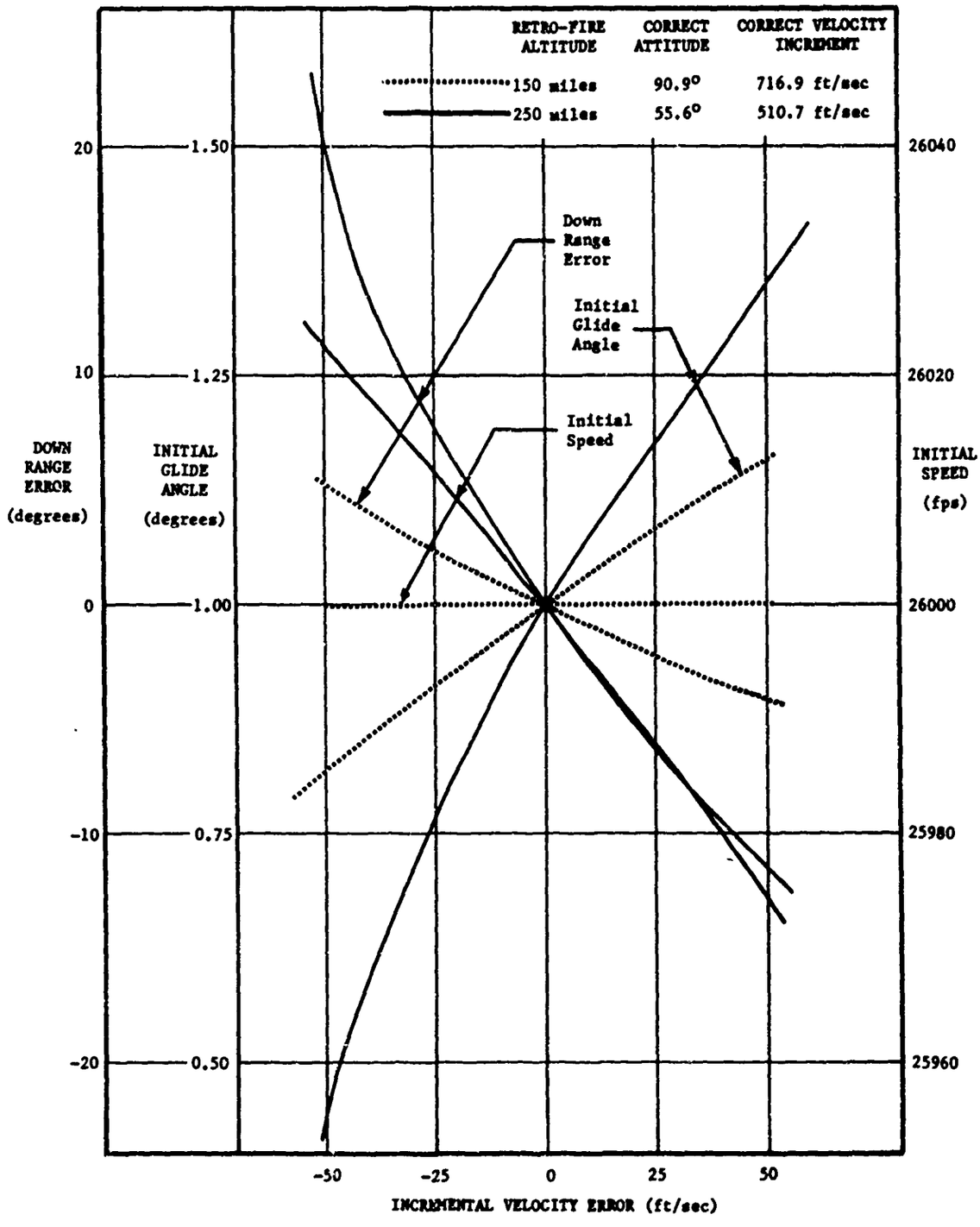
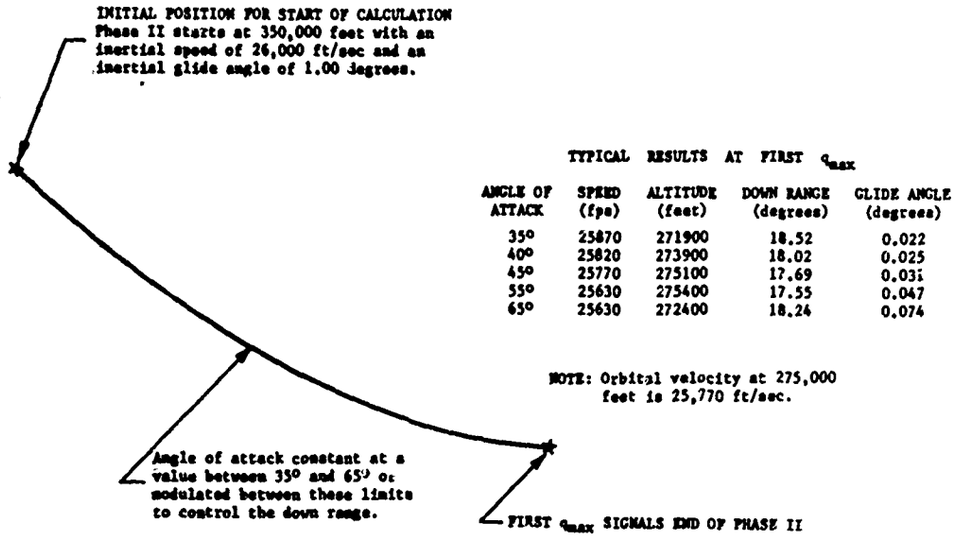


FIGURE 8b INITIAL CONDITIONS AT 350,000 FEET vs RETRO-FIRE INCREMENTAL VELOCITY ERROR FOR AN INITIALLY CIRCULAR ORBIT

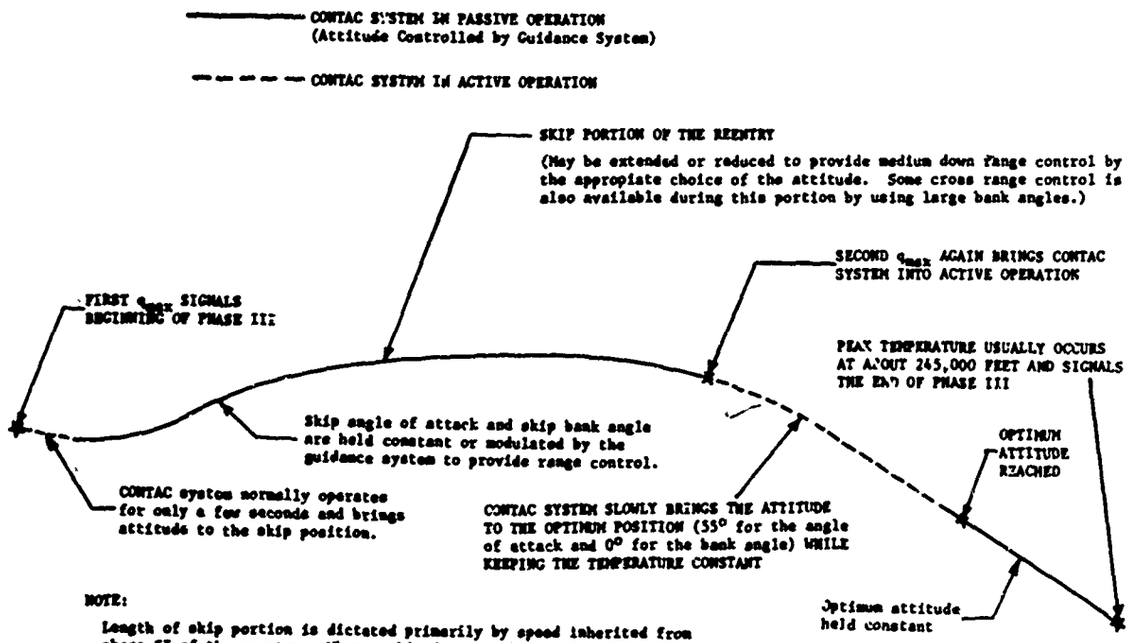


TYPICAL RESULTS AT FIRST q_{max}

ANGLE OF ATTACK	SPEED (fps)	ALTITUDE (feet)	DOWN RANGE (degrees)	GLIDE ANGLE (degrees)
35°	25870	271900	18.52	0.022
40°	25820	273900	18.02	0.025
45°	25770	275100	17.69	0.031
55°	25630	275400	17.55	0.047
65°	25630	272400	18.24	0.074

NOTE: Orbital velocity at 275,000 feet is 25,770 ft/sec.

FIGURE 9 PHASE II - OVERALL DOWN RANGE CONTROL



NOTE:

Length of skip portion is dictated primarily by speed inherited from phase II of the reentry. When combined with a low angle of attack, a speed slightly higher than orbital velocity produces extended skips up to 120 degrees of equivalent longitude. When large angles of attack and large bank angles are combined with a speed slightly less than orbital velocity, the skip portion of phase III can be eliminated completely and the total down range increment during phase III can be reduced to about 10 degrees of equivalent longitude.

FIGURE 10 PHASE III - EQUILIBRIUM GLIDE ATTAINMENT

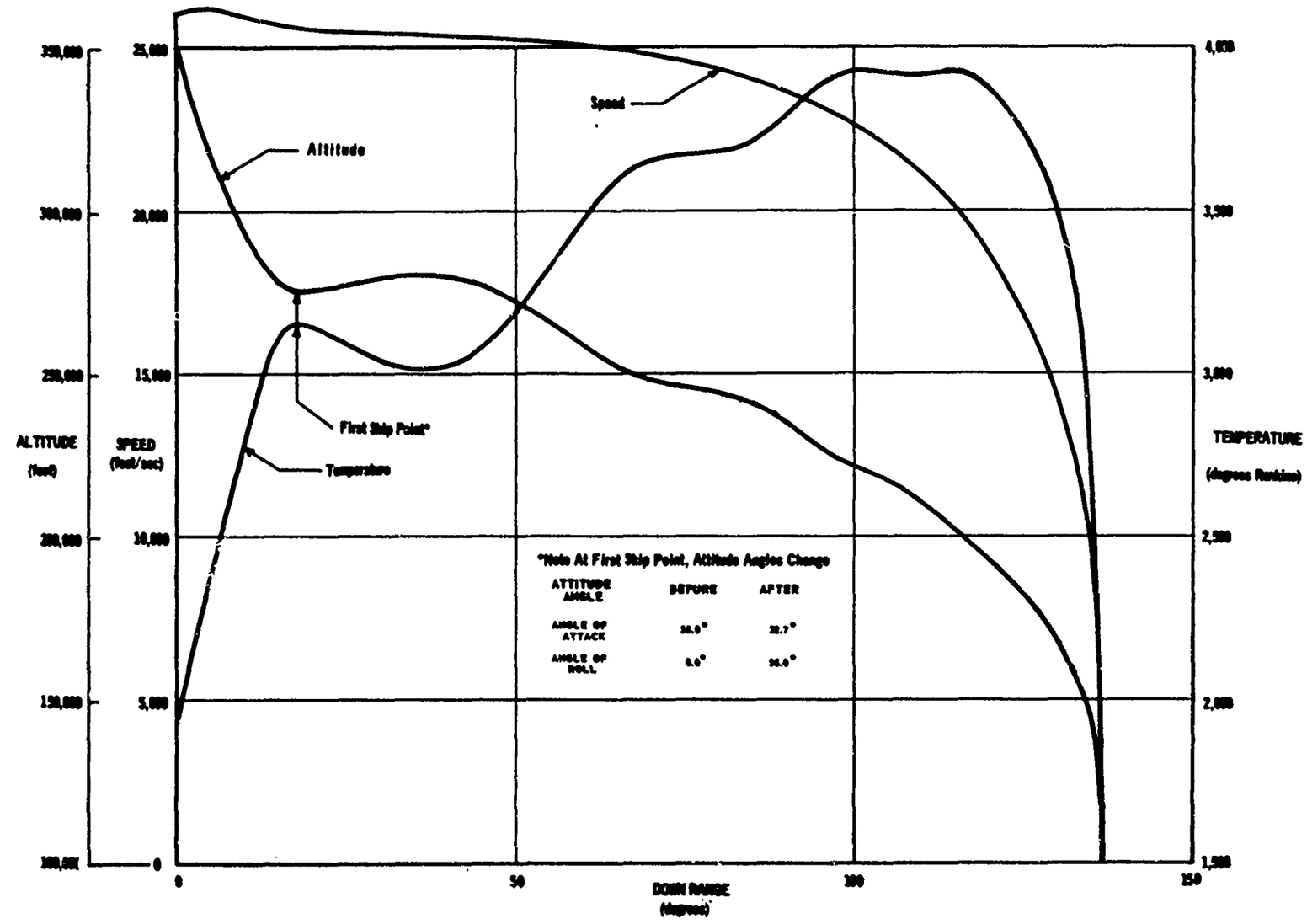


FIGURE 11 a FLIGHT PARAMETERS FOR AN UNCONTROLLED REENTRY

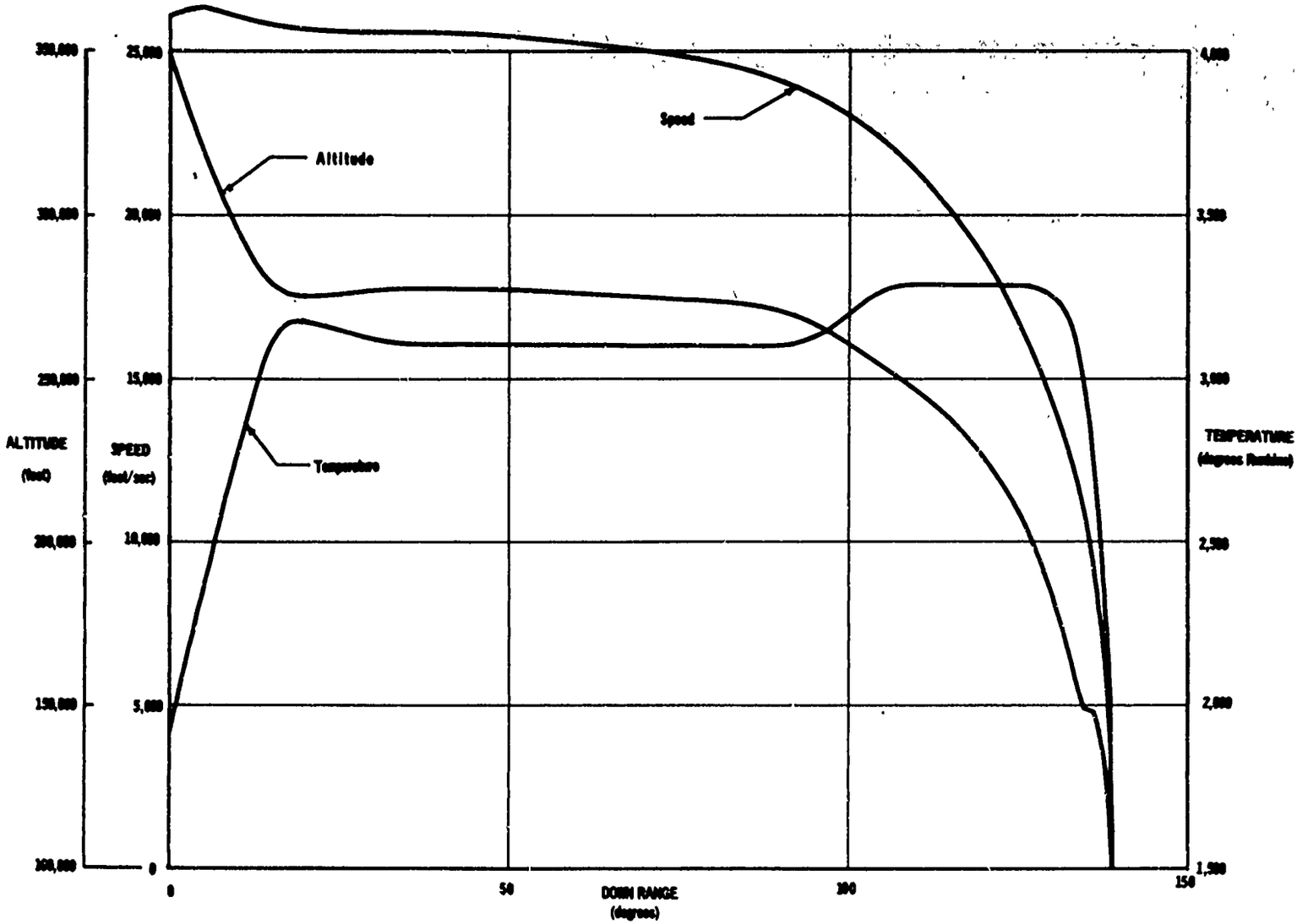


FIGURE 11b FLIGHT PARAMETERS FOR A CONTAC CONTROLLED REENTRY

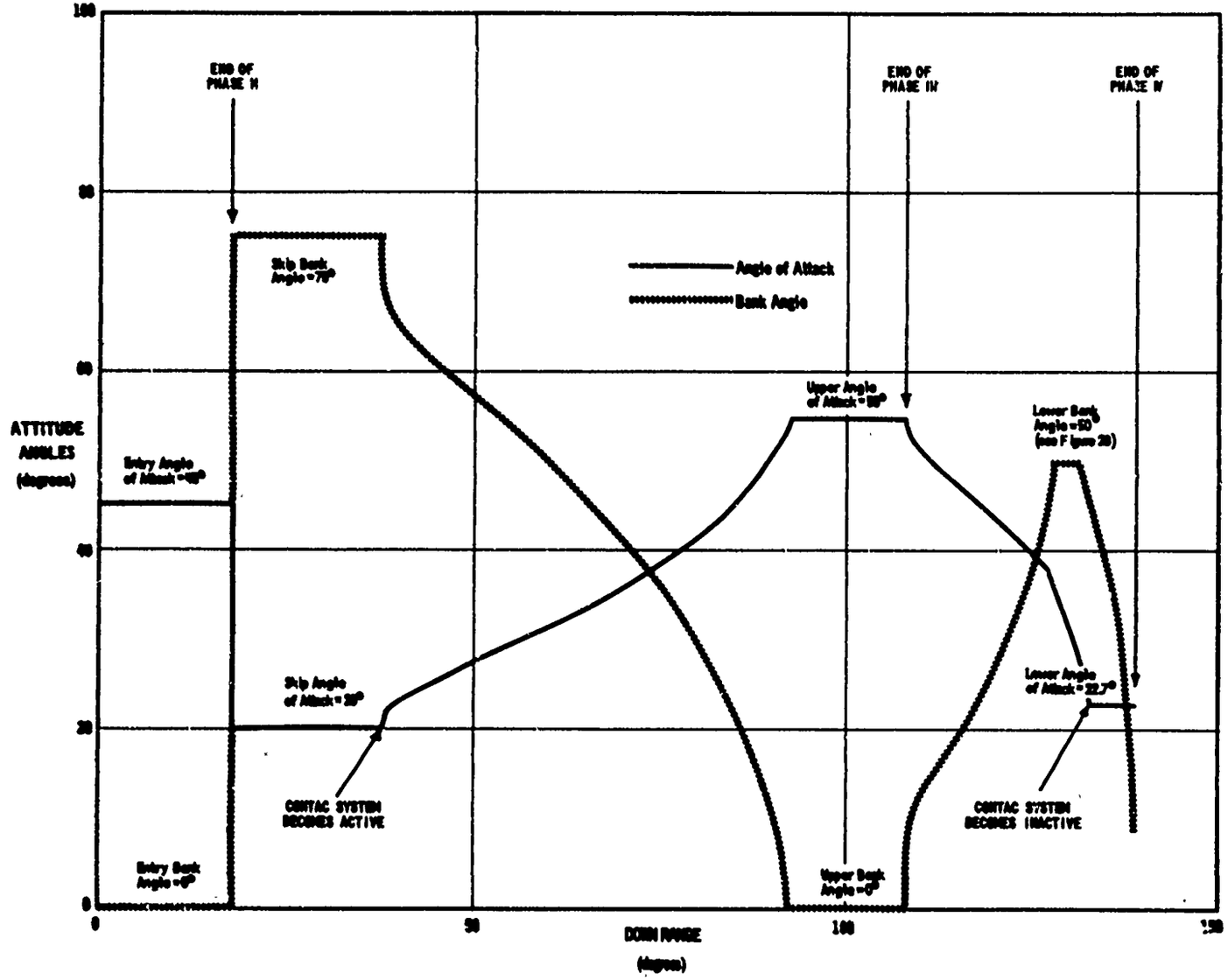


FIGURE 11c HISTORY OF ATTITUDE ANGLES FOR A CONTAC CONTROLLED REENTRY

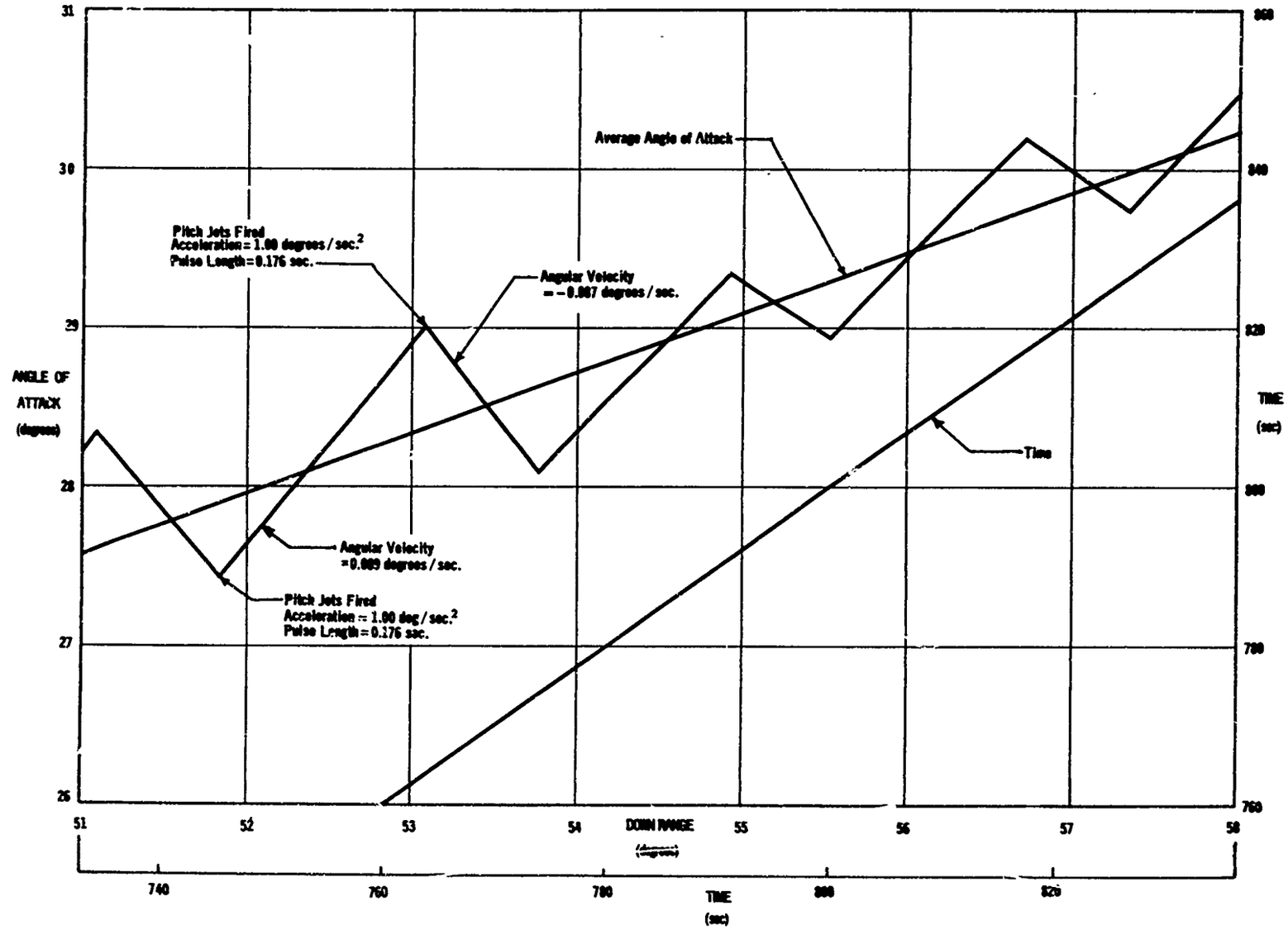


FIGURE 11d DETAILED HISTORY OF ANGLE OF ATTACK FOR A PORTION OF A CONTAC CONTROLLED REENTRY

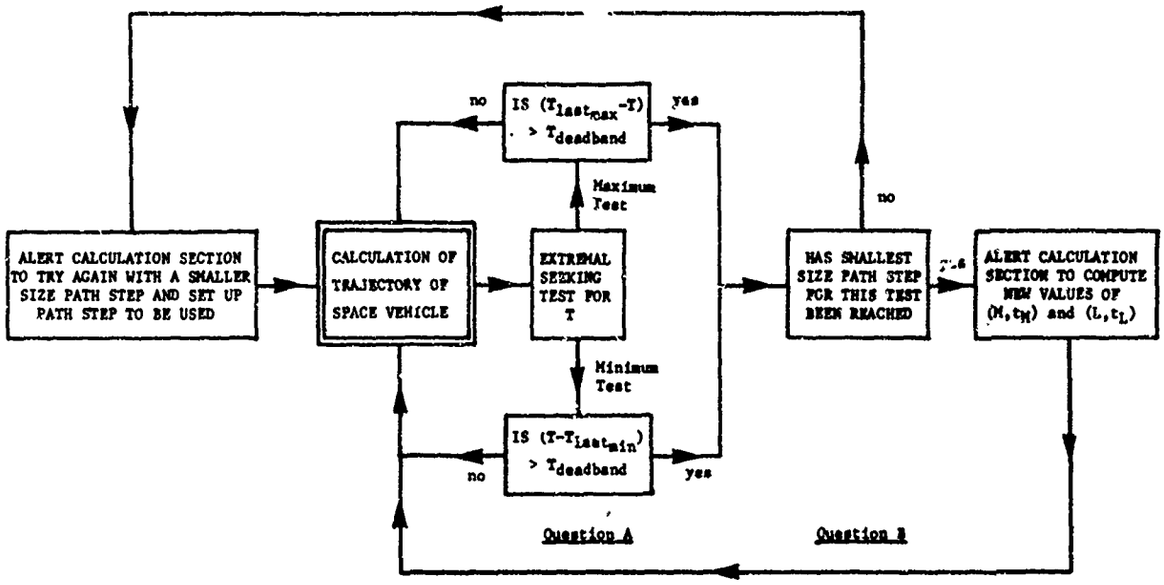


FIGURE 12 CONTAC SYSTEM PROGRAM FLOW DIAGRAM

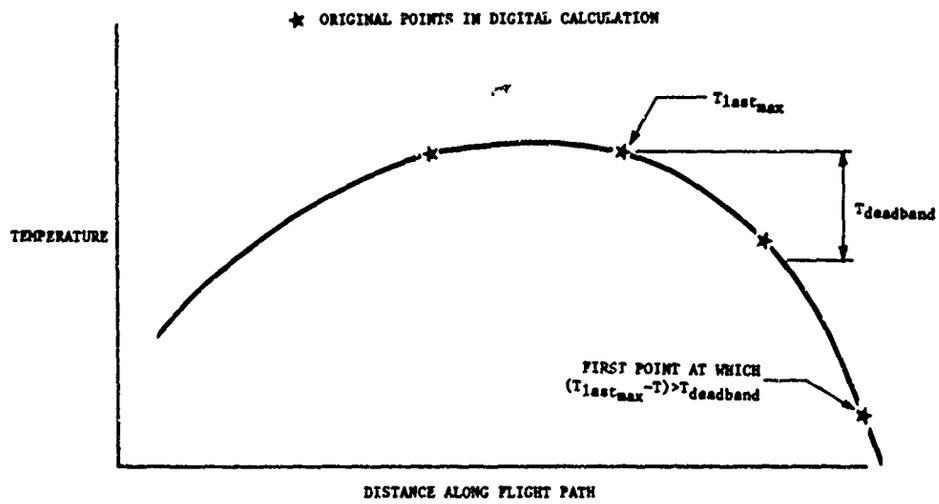


FIGURE 13 TEMPERATURE *vs* DISTANCE ALONG FLIGHT PATH
DIGITAL POINTS AND $T_{deadband}$ DIFFICULTY

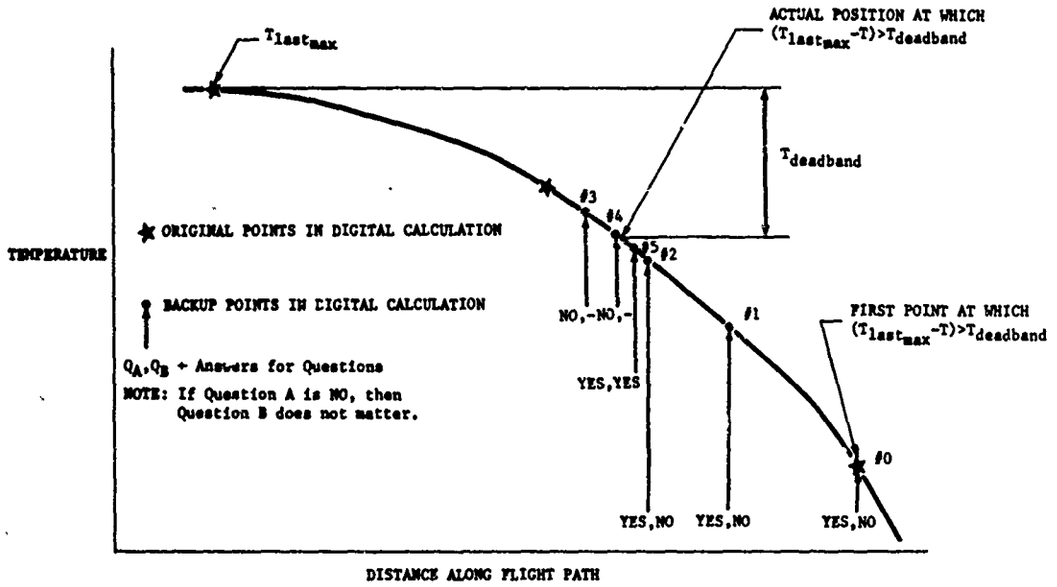


FIGURE 14 TEMPERATURE vs DISTANCE ALONG FLIGHT PATH
DIGITAL POINTS AND $T_{deadband}$ DIFFICULTY

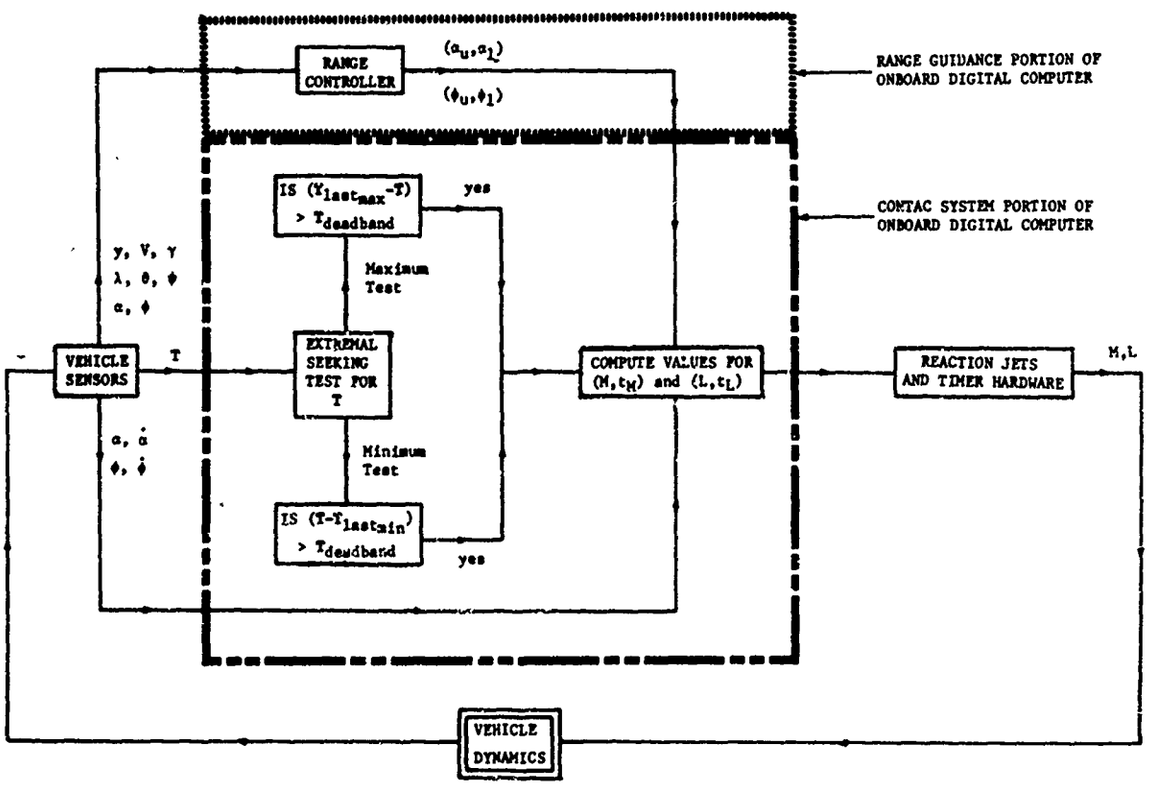


FIGURE 15 ONBOARD DIGITAL CONTAC SYSTEM FLOW DIAGRAM

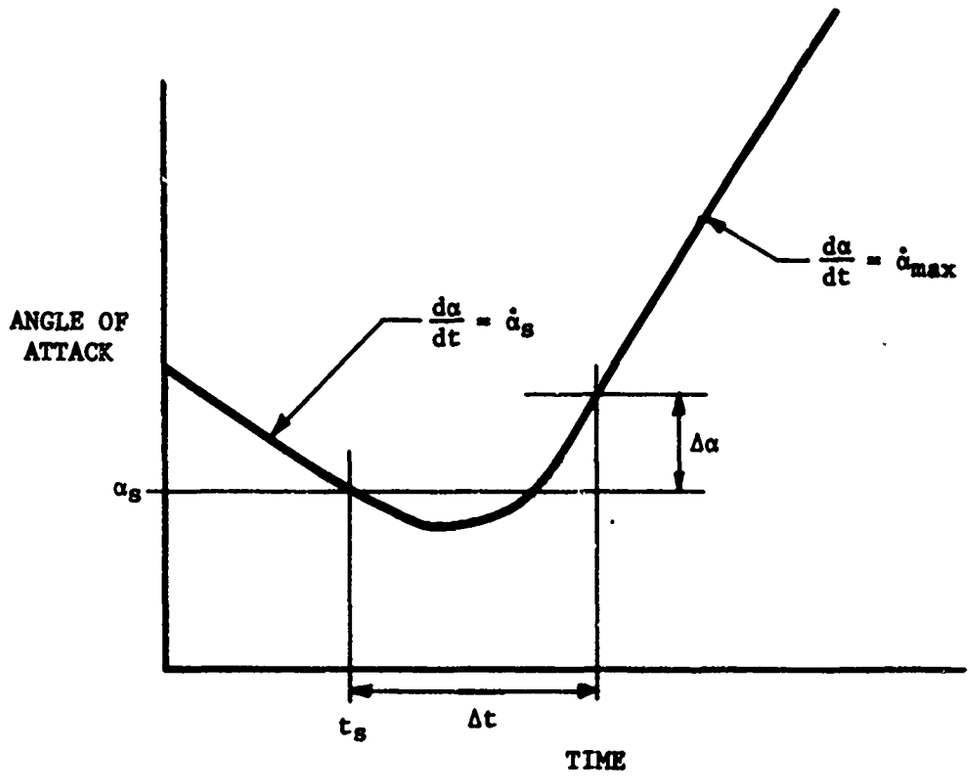


FIGURE 16 CONTAC SYSTEM CONTROL OF THE ANGLE OF ATTACK

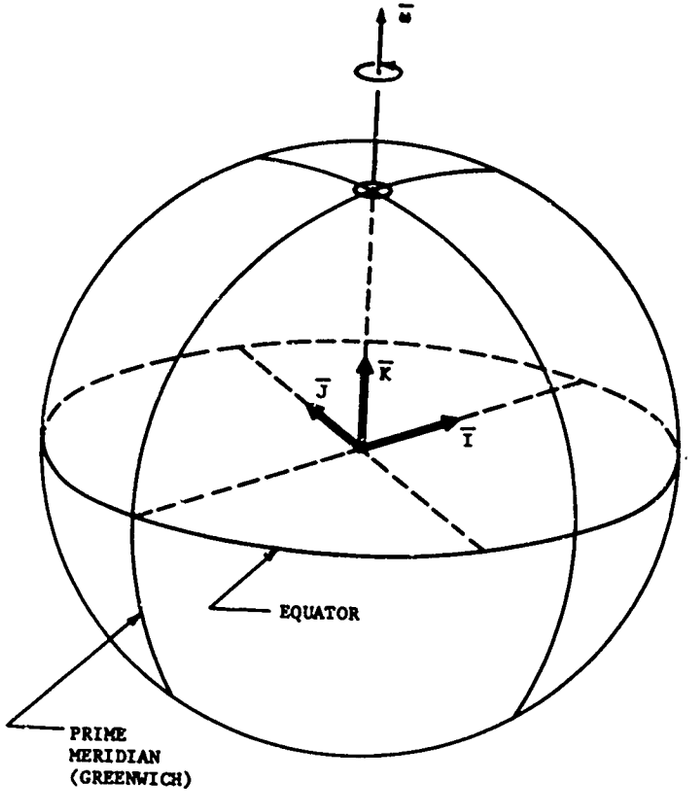


FIGURE 18 BASIC REFERENCE SYSTEM

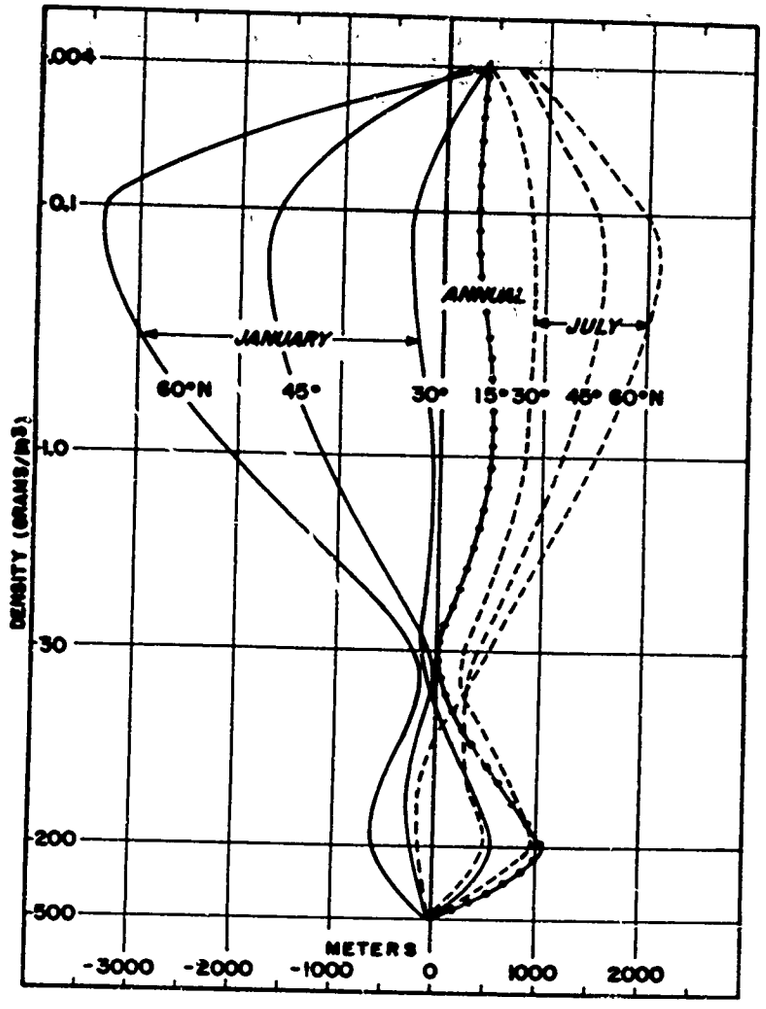


FIGURE 17 DEPARTURES IN THE HEIGHT OF SPECIFIC DENSITY SURFACES FROM U.S. STANDARD ATMOSPHERE, 1962.

(From 'Horizontal and Vertical Distribution of Atmospheric Density up to 90 km', Allen E. Cole and Arthur J. Kantor, Air Force Surveys in Geophysics, No. 157, AFRCRL–64–483, June 1964.)

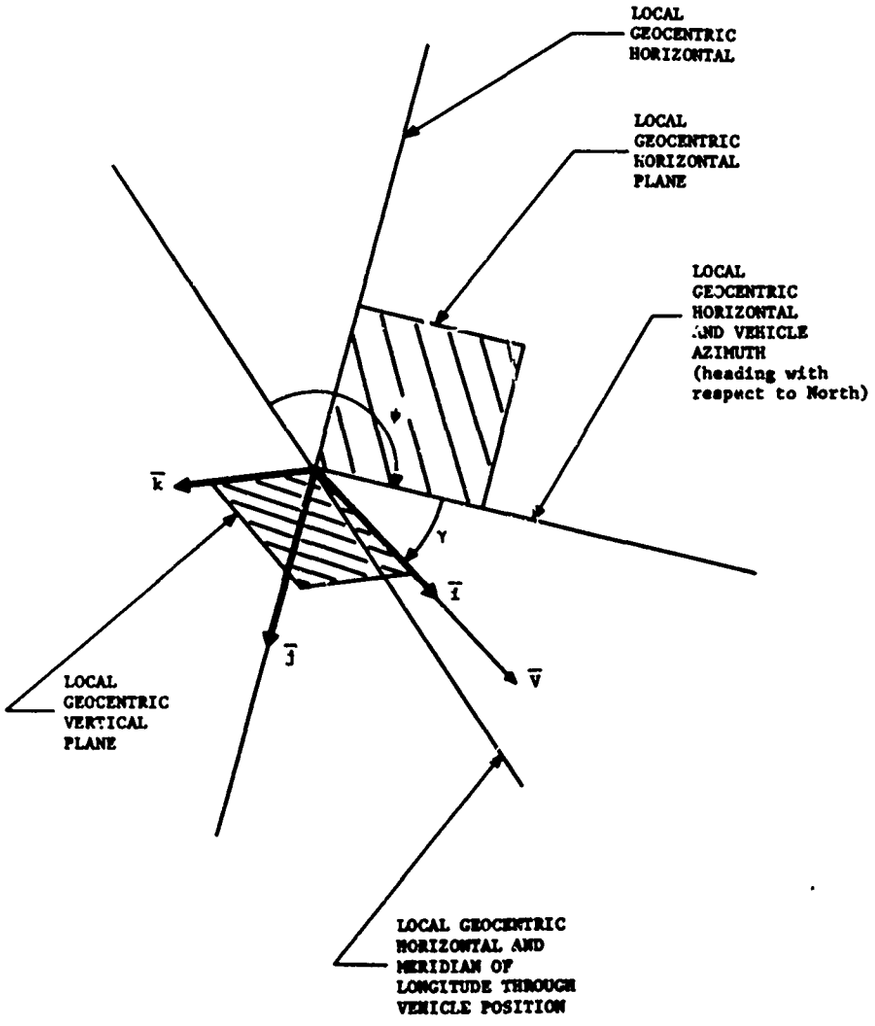


FIGURE 20 FLIGHT AXES COORDINATE REFERENCE SYSTEM

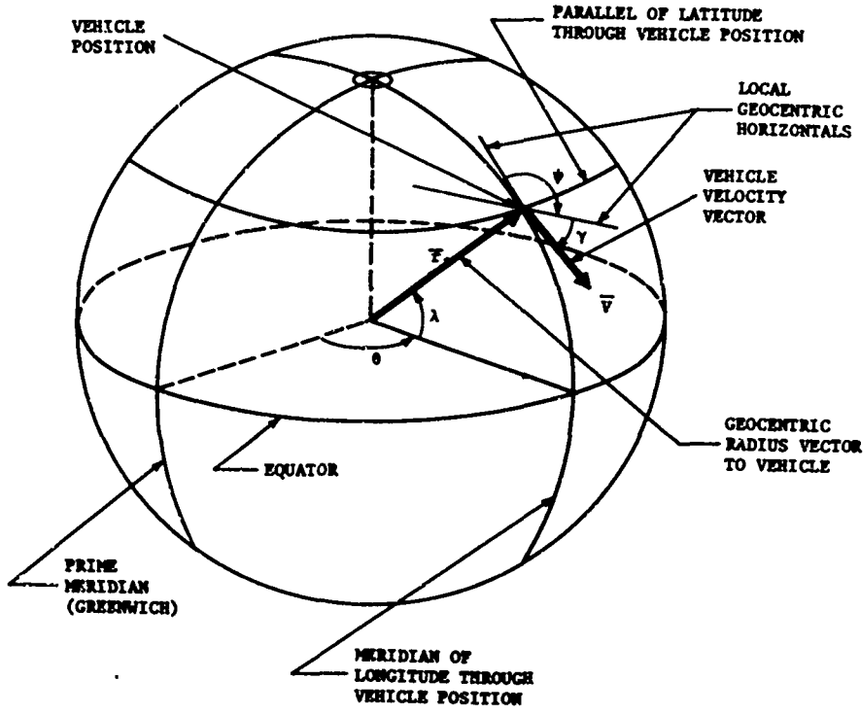
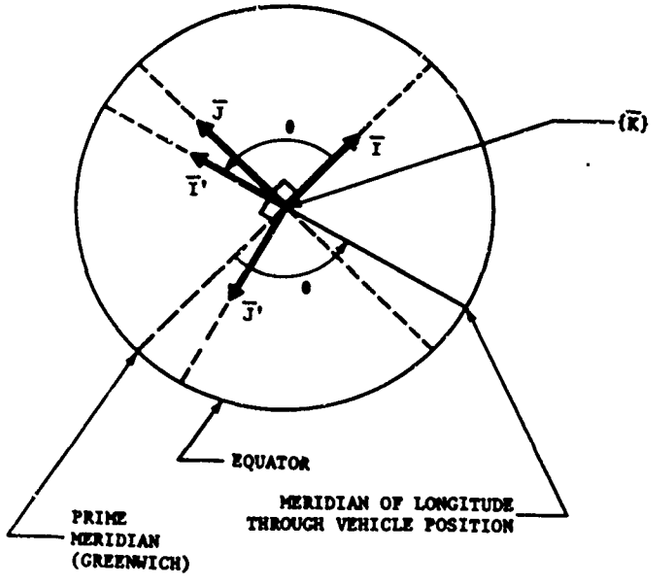


FIGURE 19 GEOCENTRIC COORDINATES FOR POSITION AND VELOCITY

NOTE:

1. $(\bar{K})$ axis is pointing out of page from indicated position



$$\begin{Bmatrix} \bar{I} \\ \bar{J} \\ \bar{K} \end{Bmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \bar{I}' \\ \bar{J}' \\ \bar{K} \end{Bmatrix}$$

$$\begin{Bmatrix} \bar{I}' \\ \bar{J}' \\ \bar{K} \end{Bmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \bar{I} \\ \bar{J} \\ \bar{K} \end{Bmatrix}$$

FIGURE 22 CONVERSION BETWEEN BASIC REFERENCE SYSTEM $\{\bar{I},\bar{J},\bar{K}\}$ AND FIRST INTERMEDIATE REFERENCE SYSTEM $\{\bar{I}',\bar{J}',\bar{K}\}$

NOTES:

1. $\bar{I}$, $\bar{J}$, $\bar{I}'$, and $\bar{J}'$ are all in the equatorial plane.
2. $\bar{n}$, $\bar{e}$, $\bar{h}$, and $\bar{j}$ are all in the local geocentric horizontal plane.
3. $\bar{h}$, $\bar{v}$, $\bar{i}$, and $\bar{k}$ are all in the same local geocentric vertical plane.

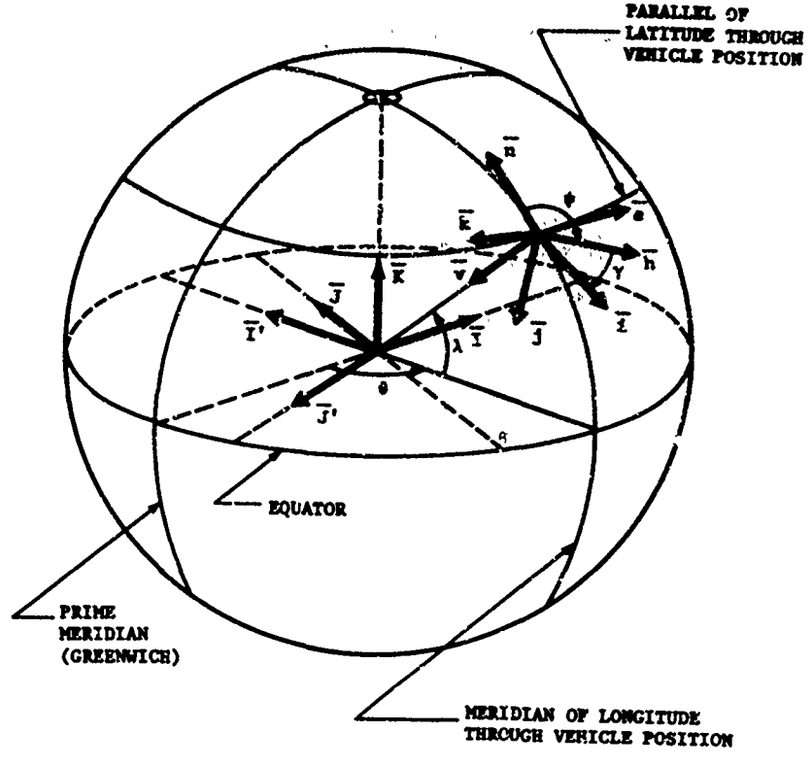


FIGURE 21 REFERENCE SYSTEMS

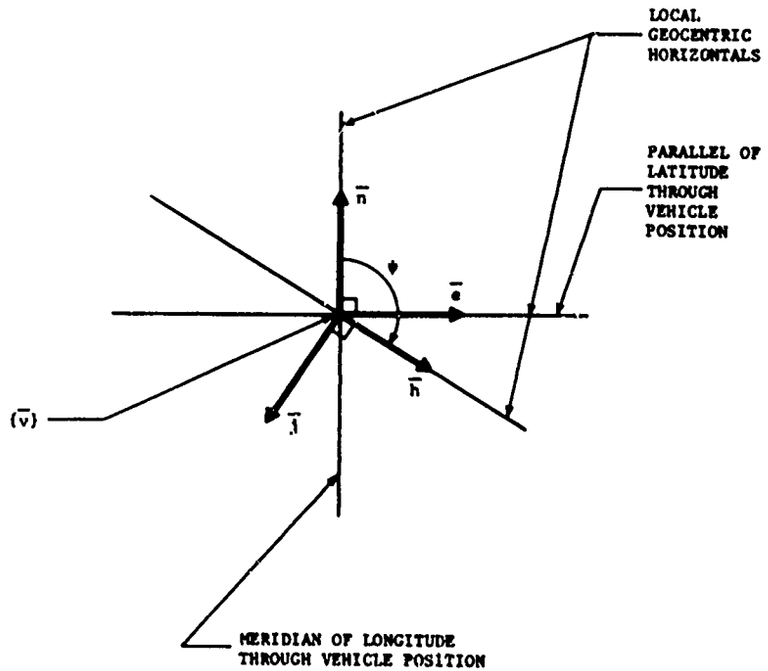
BASIC: $\{\bar{I},\bar{J},\bar{K}\}$

INTERMEDIATE: $\{\bar{I}',\bar{J}',\bar{K}\}$, $\{\bar{n},\bar{e},\bar{v}\}$, $\{\bar{h},\bar{j},\bar{v}\}$

FLIGHT: $\{\bar{i},\bar{j},\bar{k}\}$

NOTE:

1. $(\bar{v})$ axis is pointing into page at indicated position.



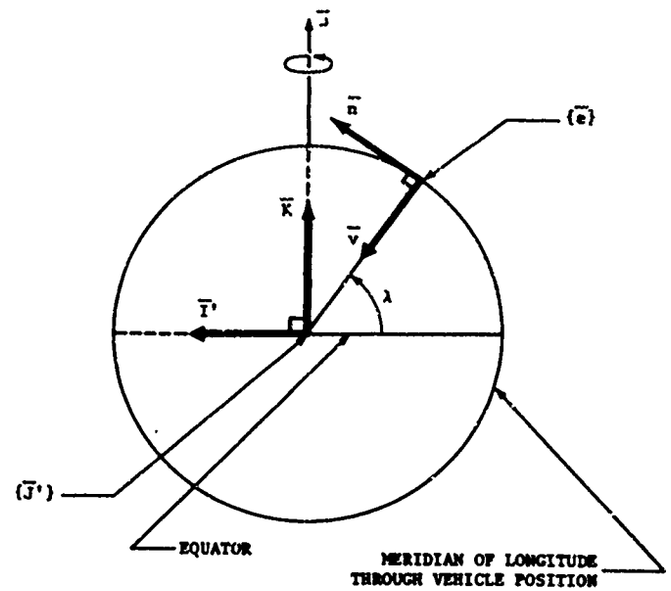
$$\begin{Bmatrix} \bar{h} \\ \bar{j} \\ \bar{v} \end{Bmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \bar{n} \\ \bar{e} \\ \bar{v} \end{Bmatrix}$$

$$\begin{Bmatrix} \bar{n} \\ \bar{e} \\ \bar{v} \end{Bmatrix} = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{Bmatrix} \bar{h} \\ \bar{j} \\ \bar{v} \end{Bmatrix}$$

FIGURE 24 CONVERSION BETWEEN SECOND INTERMEDIATE REFERENCE SYSTEM $(\bar{n},\bar{e},\bar{v})$ AND THIRD INTERMEDIATE REFERENCE SYSTEM $(\bar{h},\bar{j},\bar{v})$

NOTES:

1. $(\bar{J}')$ axis is pointing out of page from indicated position.
2. $(\bar{e})$ axis is pointing into page at indicated position.



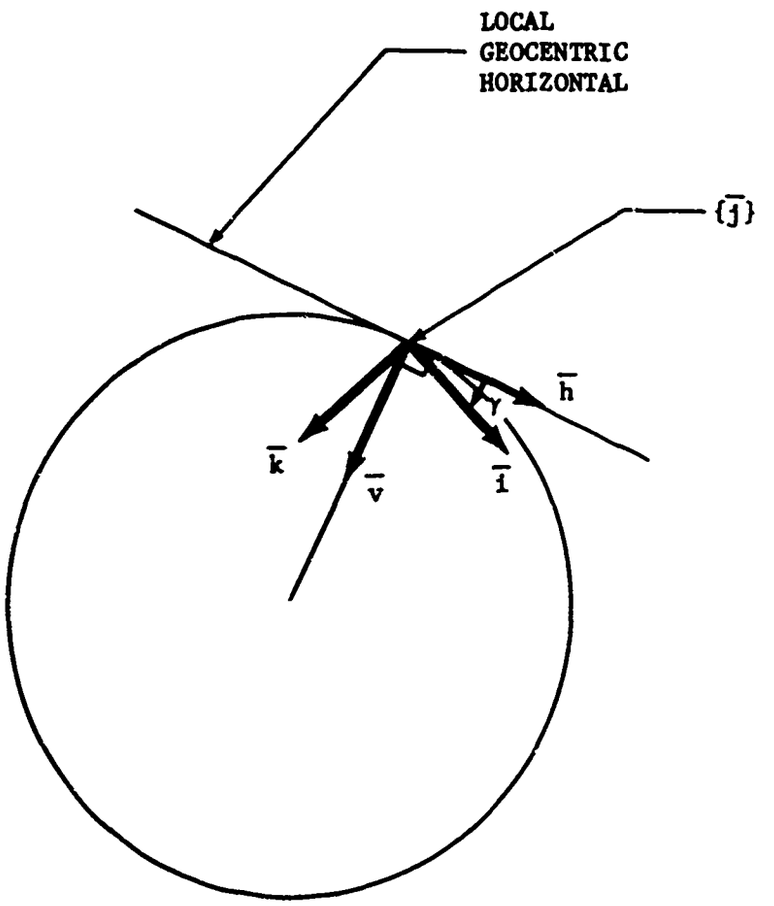
$$\begin{Bmatrix} \bar{I}' \\ \bar{J}' \\ \bar{K} \end{Bmatrix} = \begin{pmatrix} \sin\lambda & 0 & \cos\lambda \\ 0 & -1 & 0 \\ \cos\lambda & 0 & -\sin\lambda \end{pmatrix} \begin{Bmatrix} \bar{n} \\ \bar{e} \\ \bar{v} \end{Bmatrix}$$

$$\begin{Bmatrix} \bar{n} \\ \bar{e} \\ \bar{v} \end{Bmatrix} = \begin{pmatrix} \sin\lambda & 0 & \cos\lambda \\ 0 & -1 & 0 \\ \cos\lambda & 0 & -\sin\lambda \end{pmatrix} \begin{Bmatrix} \bar{I}' \\ \bar{J}' \\ \bar{K} \end{Bmatrix}$$

FIGURE 23 CONVERSION BETWEEN FIRST INTERMEDIATE REFERENCE SYSTEM $(\bar{I}',\bar{J}',\bar{K})$ AND SECOND INTERMEDIATE REFERENCE SYSTEM $(\bar{n},\bar{e},\bar{v})$

NOTE:

1. $\{\bar{j}\}$ axis is pointing out of page from indicated position



$$\begin{Bmatrix} \bar{i} \\ \bar{j} \\ \bar{k} \end{Bmatrix} = \begin{pmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{pmatrix} \begin{Bmatrix} \bar{h} \\ \bar{j} \\ \bar{v} \end{Bmatrix}$$

$$\begin{Bmatrix} \bar{h} \\ \bar{j} \\ \bar{v} \end{Bmatrix} = \begin{pmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{pmatrix} \begin{Bmatrix} \bar{i} \\ \bar{j} \\ \bar{k} \end{Bmatrix}$$

FIGURE 25 CONVERSION BETWEEN THIRD INTERMEDIATE REFERENCE SYSTEM $\{\bar{h},\bar{j},\bar{v}\}$ AND FLIGHT AXES REFERENCE SYSTEM $\{\bar{i},\bar{j},\bar{k}\}$

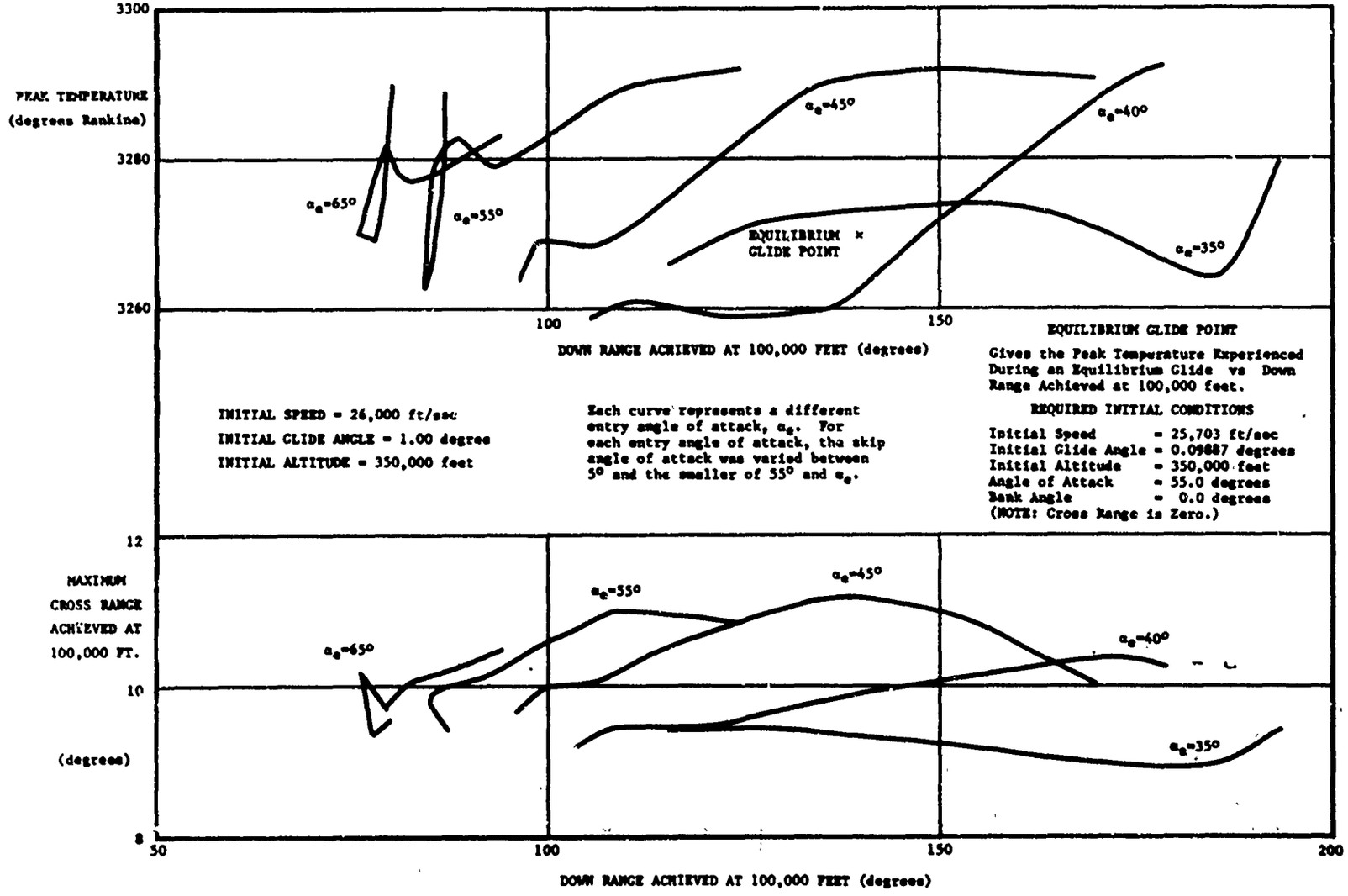


FIGURE 26 VEHICLE PERFORMANCE FOR VARIOUS ENTRY AND SKIP ANGLES OF ATTACK

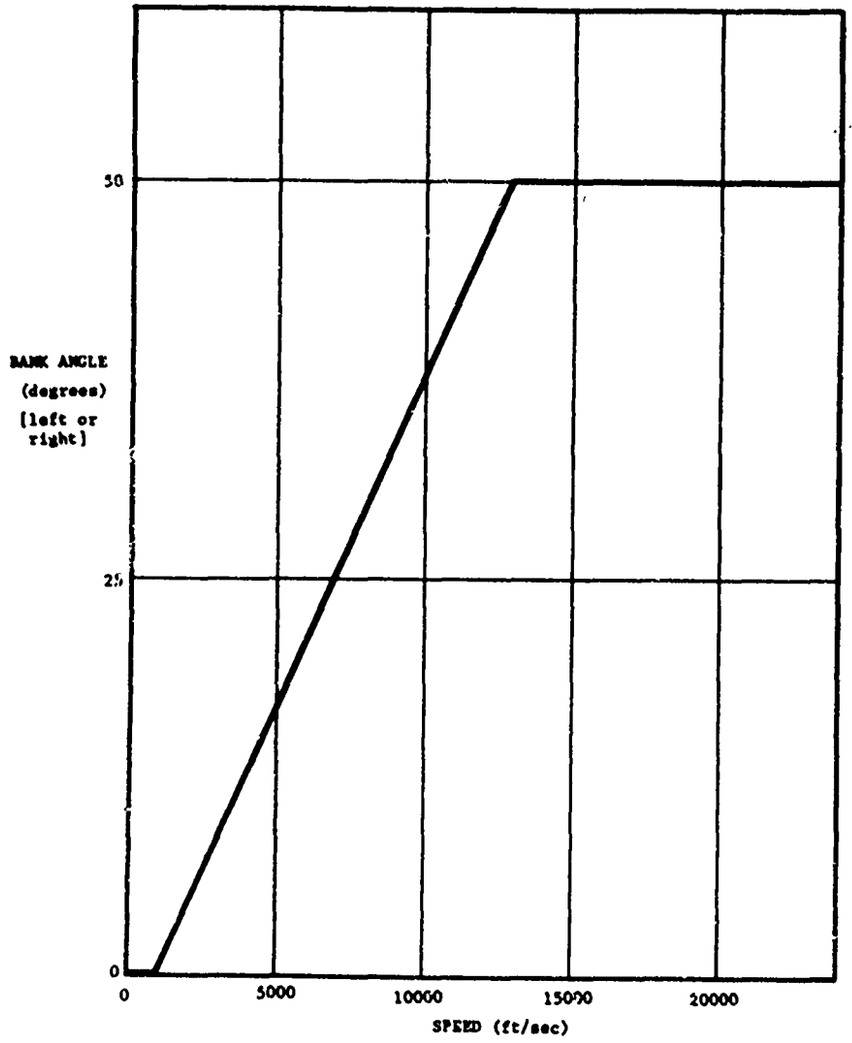


FIGURE 28 VARIATION OF LOWER LIMIT FOR THE BANK ANGLE vs SPEED DURING PHASE IV OF THE REENTRY

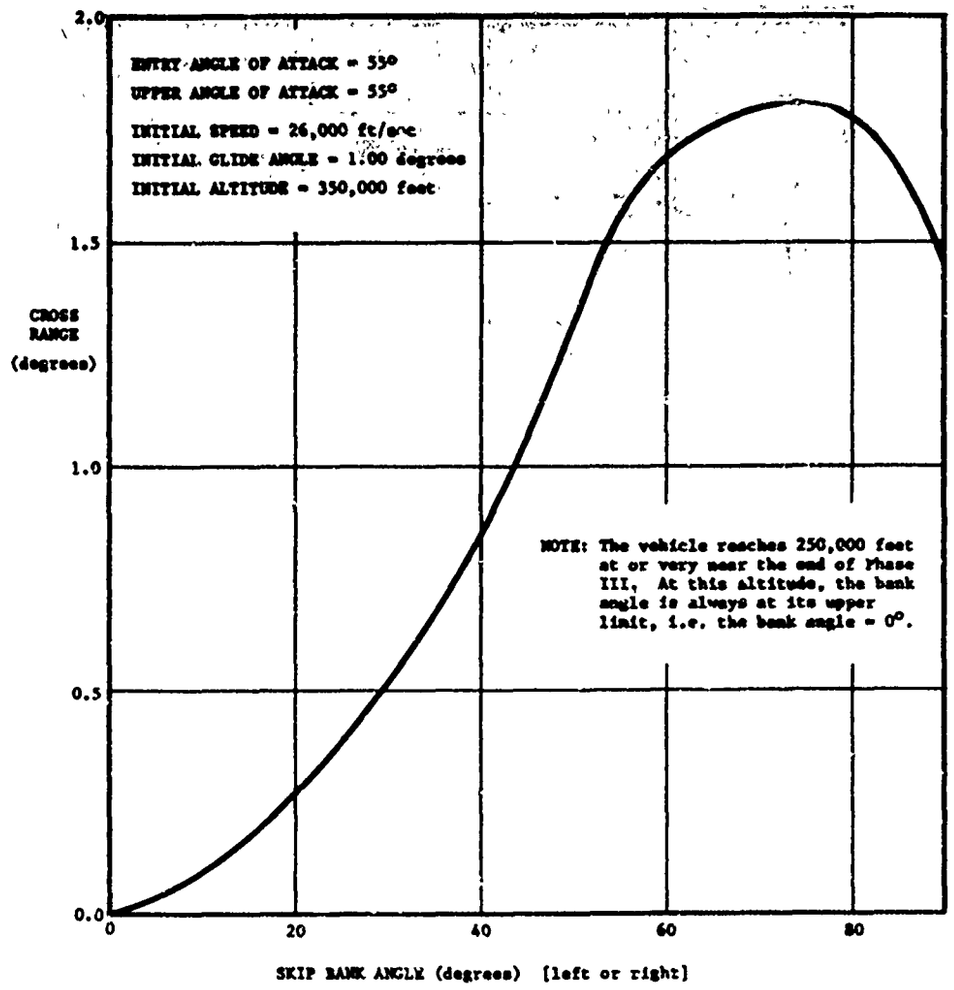
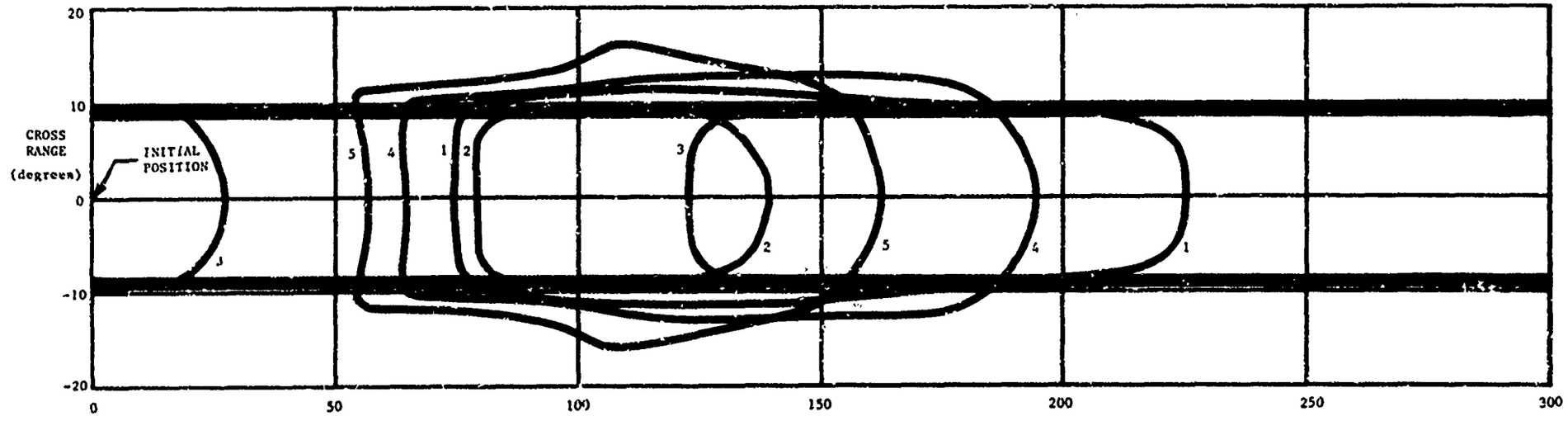


FIGURE 27 VARIATION OF CROSS RANGE AT 250,000 FEET WITH SKIP BANK ANGLE



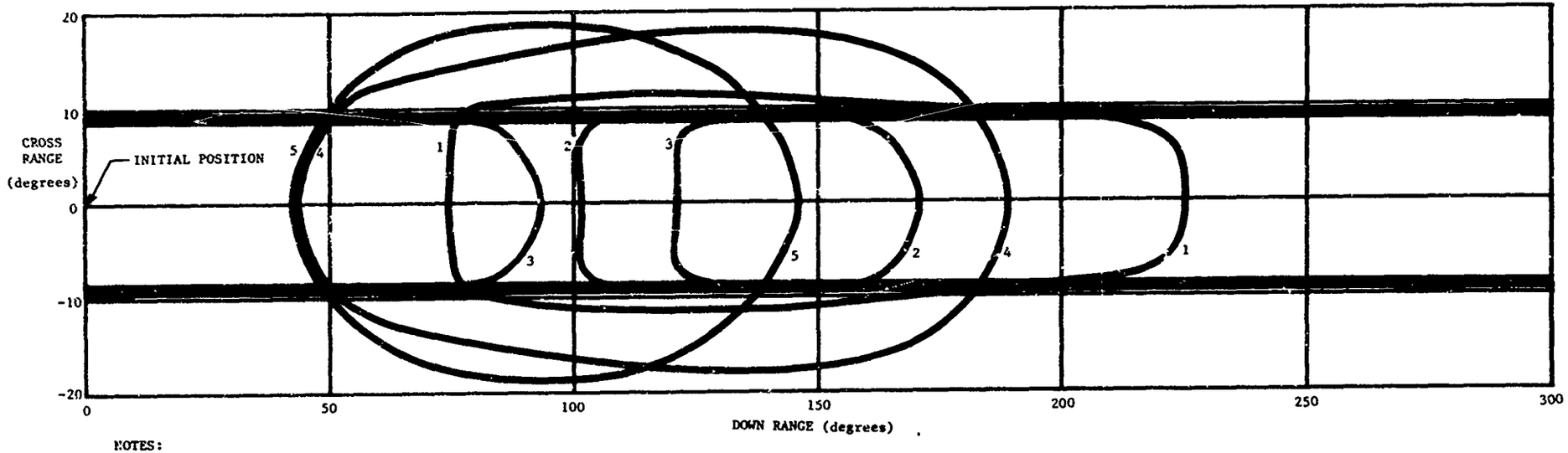
NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Conditions at the initial position are as follows:
   (a) Glide Angle = 1.00 degree
   (b) Altitude = 350,000 feet

CODE NO.	INITIAL SPEED	INITIAL ERROR IN SPEED	PEAK TEMPERATURE
1	26000 fps	0 ft/sec	2828°F
2†	26100 fps	+100 ft/sec	2828°F
3†	26300 fps	+300 ft/sec	2828°F
4	25900 fps	-100 ft/sec	3020°F
5	25700 fps	-300 ft/sec	3107°F

†Footprints 2 and 3 covered more than one complete revolution of the Earth. Consequently, the forward portion of 2 is actually at about 500 degrees down range and 3 is actually at about 155C degrees down range.

FIGURE 29 CROSS RANGE vs DOWN RANGE FOR VARIOUS INITIAL SPEEDS OVER A CIRCULAR, NON-ROTATING EARTH



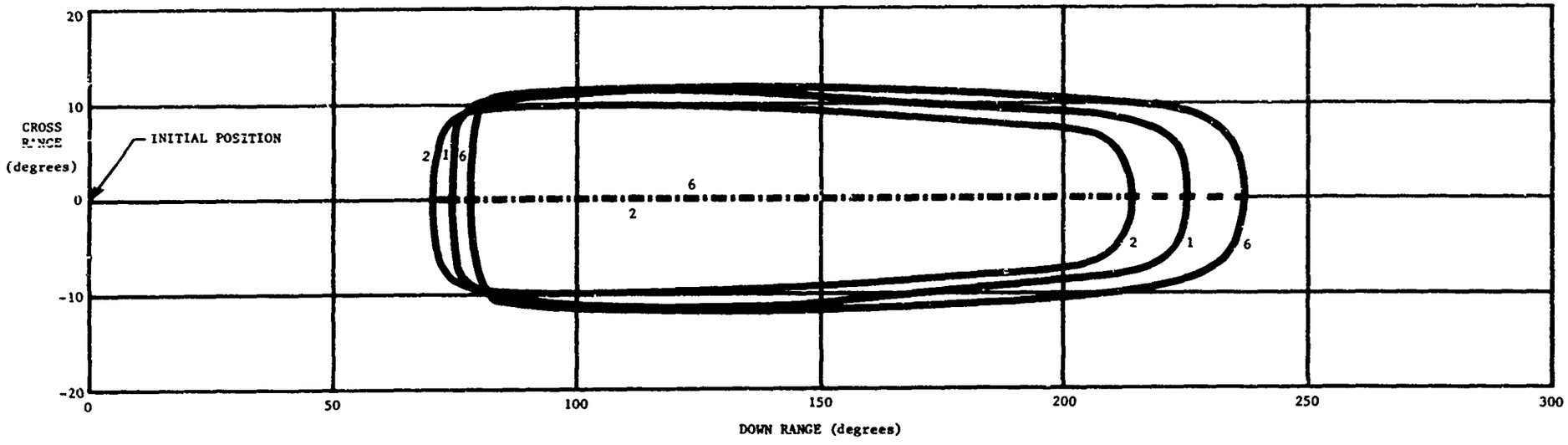
NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Conditions at the initial position are as follows:
   (a) Speed = 26,000 ft/sec
   (b) Altitude = 350,000 feet

CODE NO.	INITIAL GLIDE ANGLE	INITIAL ERROR IN GLIDE ANGLE	PEAK TEMPERATURE
1	1.00 degrees	0.00 degrees	2828°F
2†	0.75 degrees	-0.25 degrees	2828°F
3†	0.65 degrees	-0.35 degrees	2828°F
4	1.25 degrees	+0.25 degrees	2853°F
5	1.50 degrees	+0.50 degrees	3012°F

†Footprints 2 and 3 covered more than one complete revolution of the Earth. Consequently, the forward portion of 2 is actually at about 550 degrees down range and 3 is actually at about 800 degrees down range.

FIGURE 30 CROSS RANGE vs DOWN RANGE FOR VARIOUS INITIAL GLIDE ANGLES OVER A CIRCULAR, NON-ROTATING EARTH

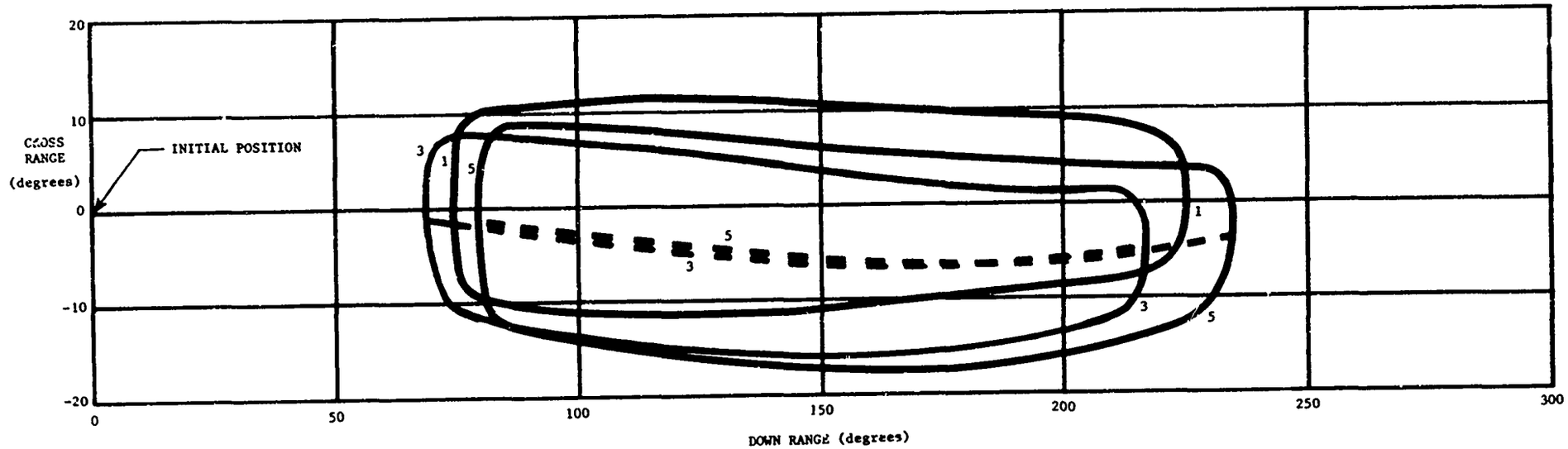


CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
1	Non-Rotating Earth Ref.		2828°F
2	0°	90°	2700°F
6	0°	-90°	2954°F

NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Inertial conditions (relative to inertial space) at the initial position are as follows:
   (a) Inertial Speed = 26,000 ft/sec
   (b) Inertial Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet
   (d) Inertial Azimuth is value given in the table; no correction is necessary for these two reentries.
4. Solid lines are the boundaries of the footprints.
5. Broken lines are the midlines of the footprints. They show the extent of the shift in cross range due to the Earth's rotation.

FIGURE 31 a CROSS RANGE vs DOWN RANGE FOR TWO INITIAL POSITIONS OVER A CIRCULAR, ROTATING EARTH

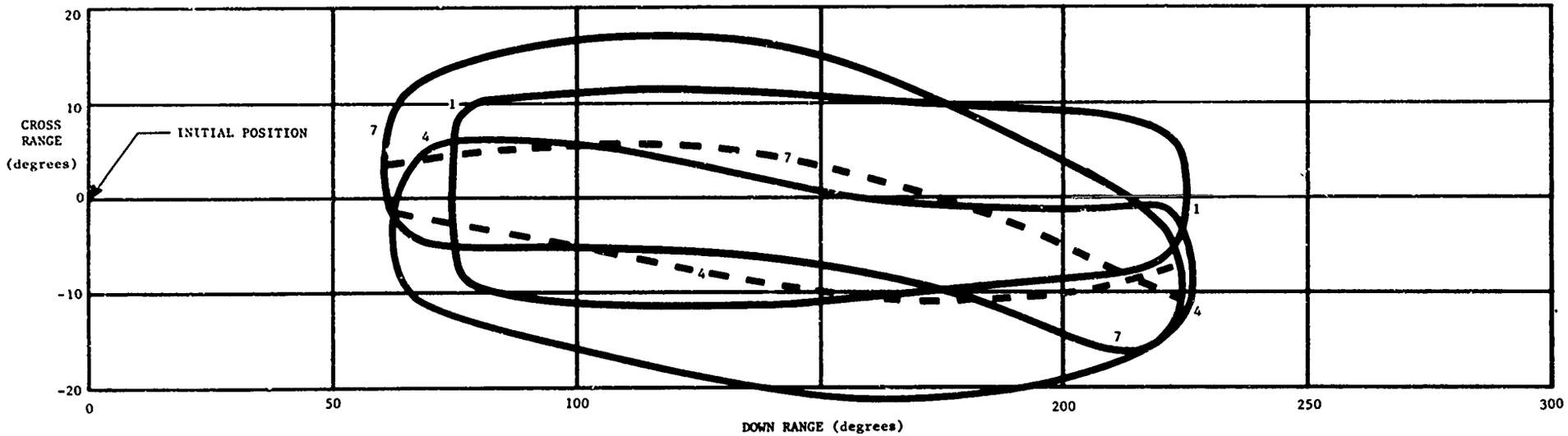


CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
1	Non-Rotating Earth Ref.		2828°F
3	0°	55°	2716°F
5	0°	-55°	2934°F

NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,900 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Inertial conditions (relative to inertial space) at the initial position are as follows:
   (a) Inertial Speed = 26,000 ft/sec
   (b) Inertial Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet
   (d) Inertial Azimuth is value given in table; a correction must be applied (-2.0 degrees) to obtain the azimuth relative to a rotating Earth.
4. Solid lines are the boundaries of the footprints.
5. Broken lines are the midlines of the footprints. They show the extent of the shift in cross range due to the Earth's rotation.

FIGURE 31b CROSS RANGE vs DOWN RANGE FOR TWO INITIAL POSITIONS OVER A CIRCULAR, ROTATING EARTH

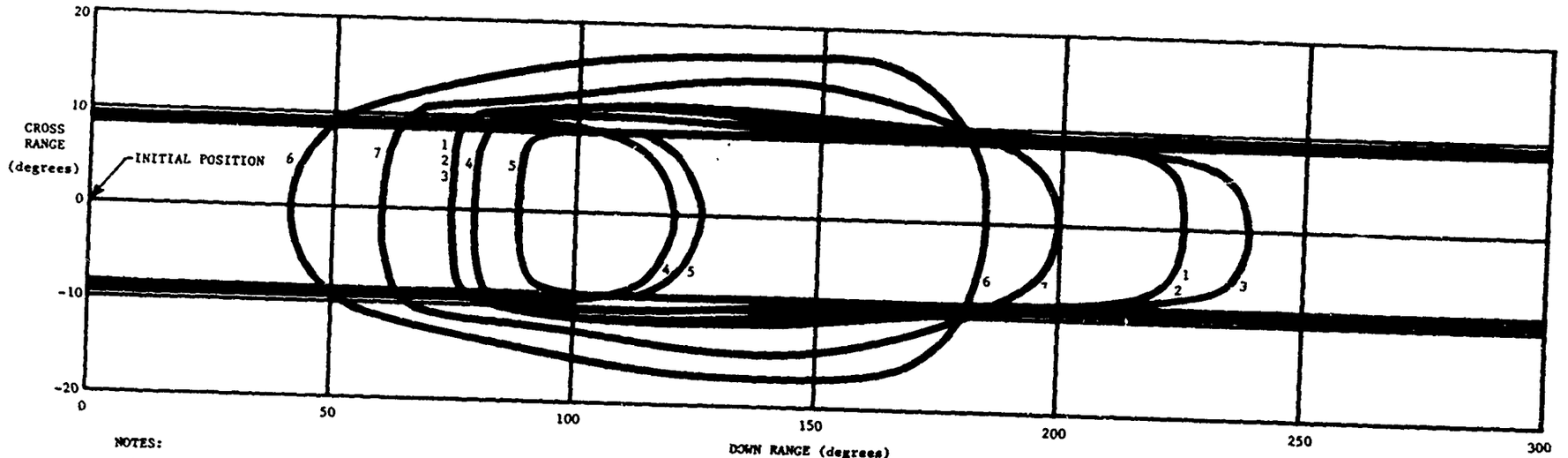


CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
1	Non-Rotating Earth Ref.		2828°F
4	0°	0°	2833°F
7	-90°	0°	2835°F

NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Inertial conditions (relative to inertial space) at the initial position are as follows:
   (a) Inertial Speed = 26,000 ft/sec
   (b) Inertial Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet
   (d) Inertial Azimuth is value given in table: the azimuth for footprint 4 is -3.4 degrees relative to a rotating Earth; no correction is necessary for footprint 7 since the relative rotational speed at the pole is zero.
4. Solid lines are the boundaries of the footprints.
5. Broken lines are the midlines of the footprints. They show the extent of the shift in cross range due to the Earth's rotation

FIGURE 31c CROSS RANGE vs DOWN RANGE FOR TWO INITIAL POSITIONS OVER A CIRCULAR, ROTATING EARTH



NOTES:

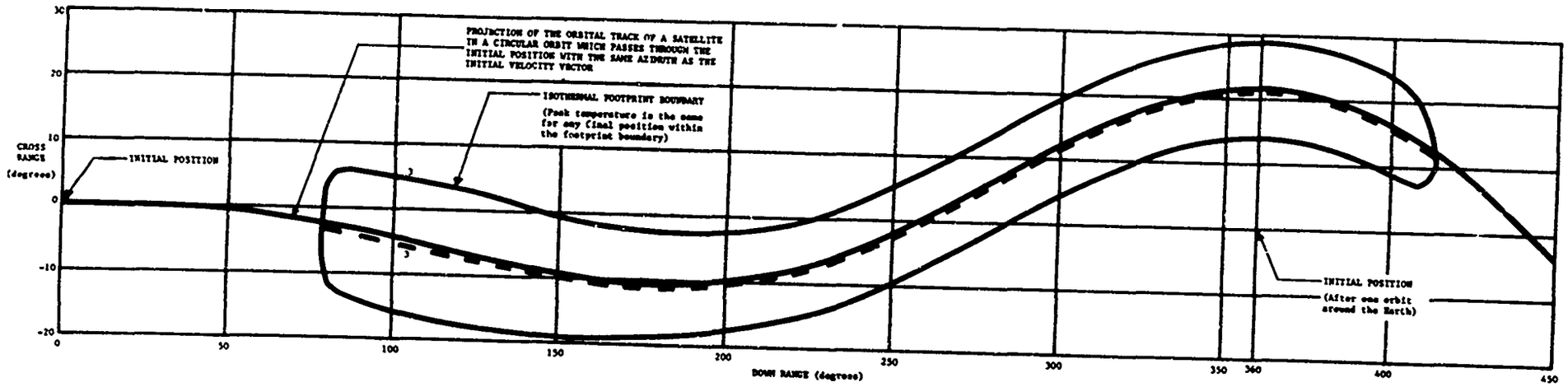
1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Conditions at the initial position are as follows:
   (a) Speed = 26,000 ft/sec
   (b) Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet (perpendicular height above the Earth's oblate surface)

CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
1	Circular Earth Ref.		2828°F
2*	0°	90°	2826°F
3*	0°	55°	2832°F
4†	0°	0°	2832°F
5†	30°	0°	2839°F
6	-60°	0°	2984°F
7	-90°	0°	2957°F

*All of footprint 2 and the rear 30 degrees of footprint 3 coincided with footprint 1.

†Footprints 4 and 5 covered a little more than one complete revolution of the Earth. Consequently, the forward portions of both footprints are actually at about 500 degrees down range.

FIGURE 32 CROSS RANGE vs DOWN RANGE FOR VARIOUS INITIAL POSITIONS AND HEADINGS OVER AN OBLATE, NON-ROTATING EARTH

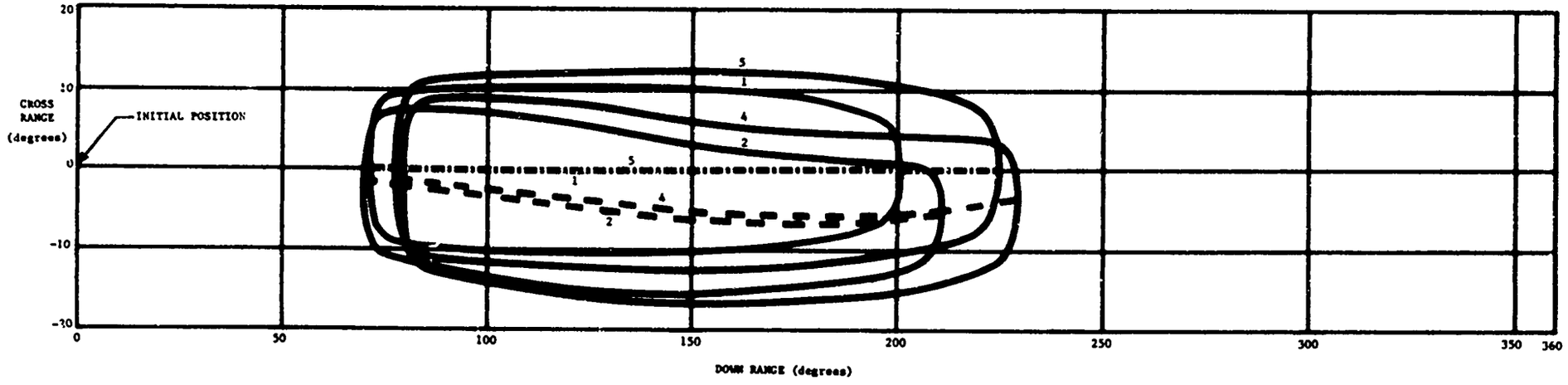


CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
3	0°	0°	2831°F

NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Inertial conditions (relative to inertial space) at the initial position are as follows:
   (a) Inertial Speed = 24,000 ft/sec
   (b) Inertial Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet (perpendicular height above Earth's oblate surface)
   (d) Inertial Azimuth is value given in table; the azimuth for footprint 3 is -3.4 degrees relative to a rotating Earth.
4. Except where otherwise noted, the solid lines are the boundaries of the footprint.
5. The broken line is the midline of footprint 3. It shows the amount of the shift in cross range due to the Earth's rotation.

FIGURE 33 — COMPARISON BETWEEN THE CROSS RANGE SHIFT OF A FOOTPRINT BOUNDARY AND THE CROSS RANGE SHIFT OF THE ORBITAL TRACK OF A SATELLITE — OBLATE, ROTATING EARTH

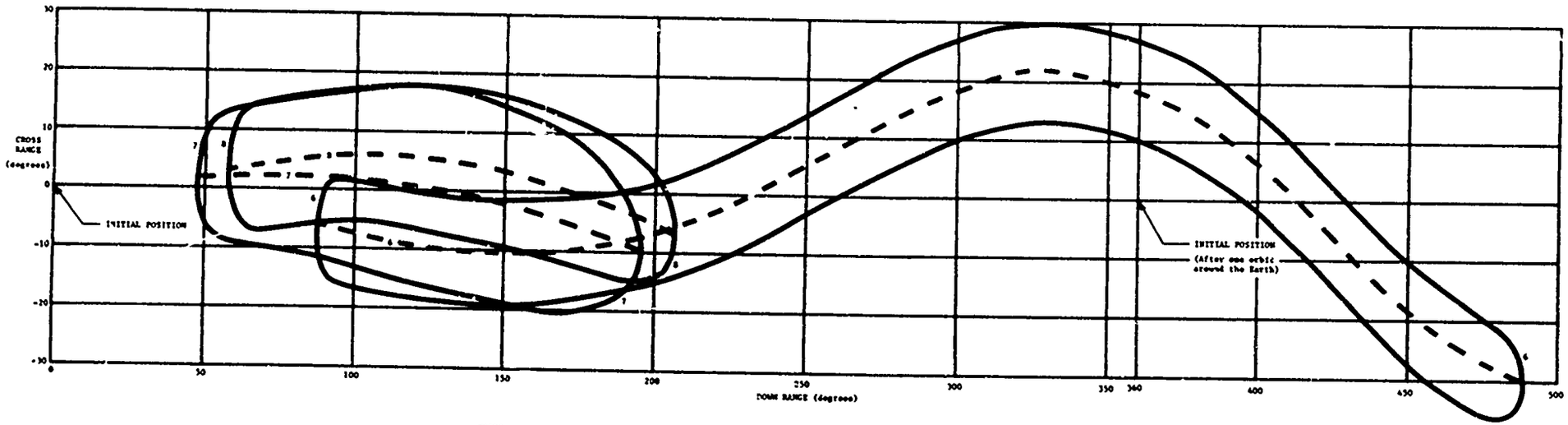


CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
1	0°	90°	2706°F
2	0°	55°	2730°F
4	0°	-55°	2931°F
5	0°	-90°	2936°F

NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 3,000 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Inertial conditions (relative to inertial space) at the initial position are as follows:
   (a) Inertial Speed = 26,000 ft/sec
   (b) Inertial Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet (perpendicular height above Earth's oblate surface)
   (d) Inertial Azimuth is value given in table; no correction is necessary for footprints 1 and 5; a correction must be applied (-2.0 degrees) for footprints 2 and 4 to obtain the azimuth relative to a rotating Earth.
4. Solid lines are the boundaries of the footprints.
5. Broken lines are the midlines of the footprints. They show the extent of the shift in cross range due to the Earth's rotation.

FIGURE 33 b CROSS RANGE vs DOWN RANGE FOR FOUR INITIAL HEADINGS OVER AN OBLATE, ROTATING EARTH



CODE NO.	INITIAL LATITUDE	INITIAL AZIMUTH	PEAK TEMPERATURE
6	30°	0°	2036°F
7	-60°	0°	2719°F
8	-90°	0°	2853°F

NOTES:

1. All positions of down range and cross range are given in degrees of equivalent longitude and latitude, respectively, relative to the great circle through the projection of the initial velocity vector.
2. Conditions on the footprint boundary are as follows:
   (a) Energy of Vehicle = 250,000 ft lb_f / lb_m
   (Nominal Speed = 2,008 ft/sec)
   (Nominal Altitude = 100,000 feet)
   (b) Heading is approximately perpendicular to the footprint boundary (±20 degrees)
3. Inertial conditions (relative to inertial space) at the initial position are as follows:
   (a) Inertial Speed = 26,000 ft/sec
   (b) Inertial Glide Angle = 1.00 degree
   (c) Altitude = 350,000 feet (perpendicular height above Earth's oblate surface)
   (d) Inertial Azimuth is value given in table; a correction must be applied (up to - 3.0 degrees) to obtain the azimuth relative to a rotating North.
4. Solid lines are the boundaries of the footprints
5. Broken lines are the midlines of the footprints. They show the extent of the shift in cross range due to the Earth's rotation.

FIGURE 33c CROSS RANGE vs DOWN RANGE FOR THREE INITIAL POSITIONS OVER AN OBLATE, ROTATING EARTH

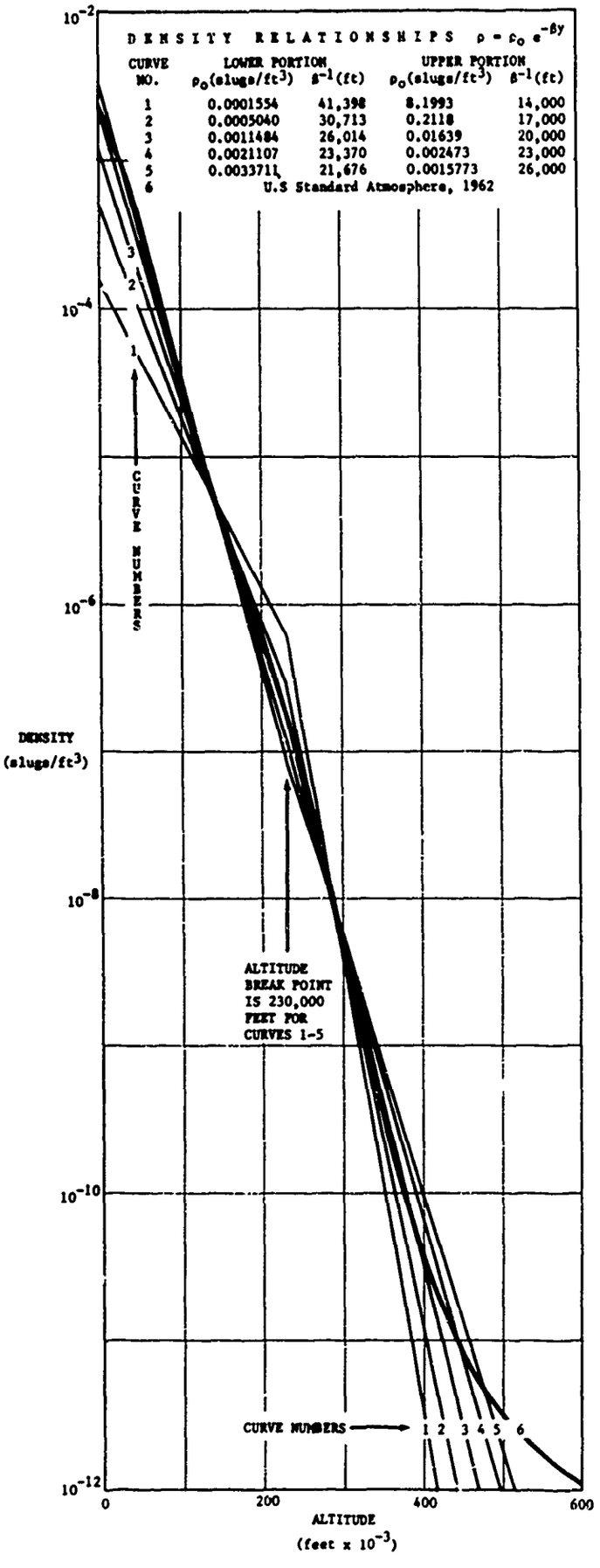


FIGURE 34 DENSITY ALTITUDE RELATIONSHIPS USED TO TEST THE PERFORMANCE OF THE CONTAC SYSTEM

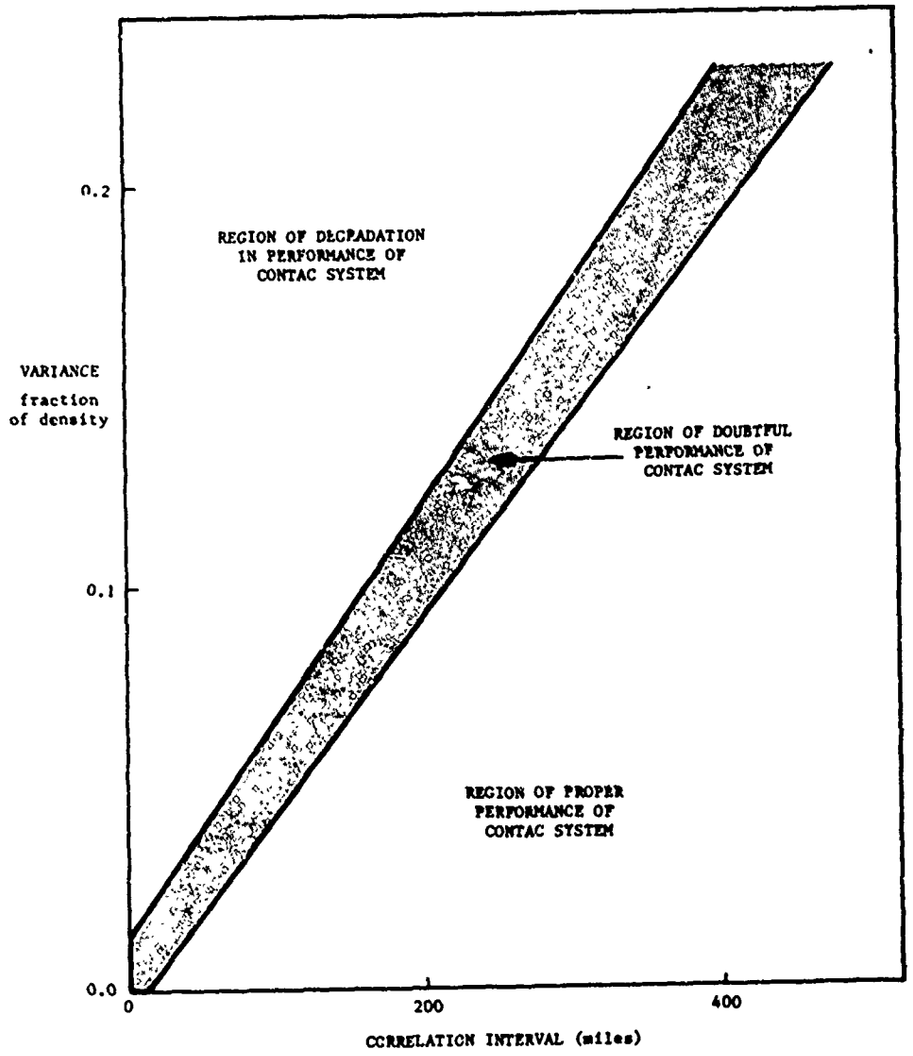


FIGURE 36 OPERATIONAL REGION OF CONTAC SYSTEM

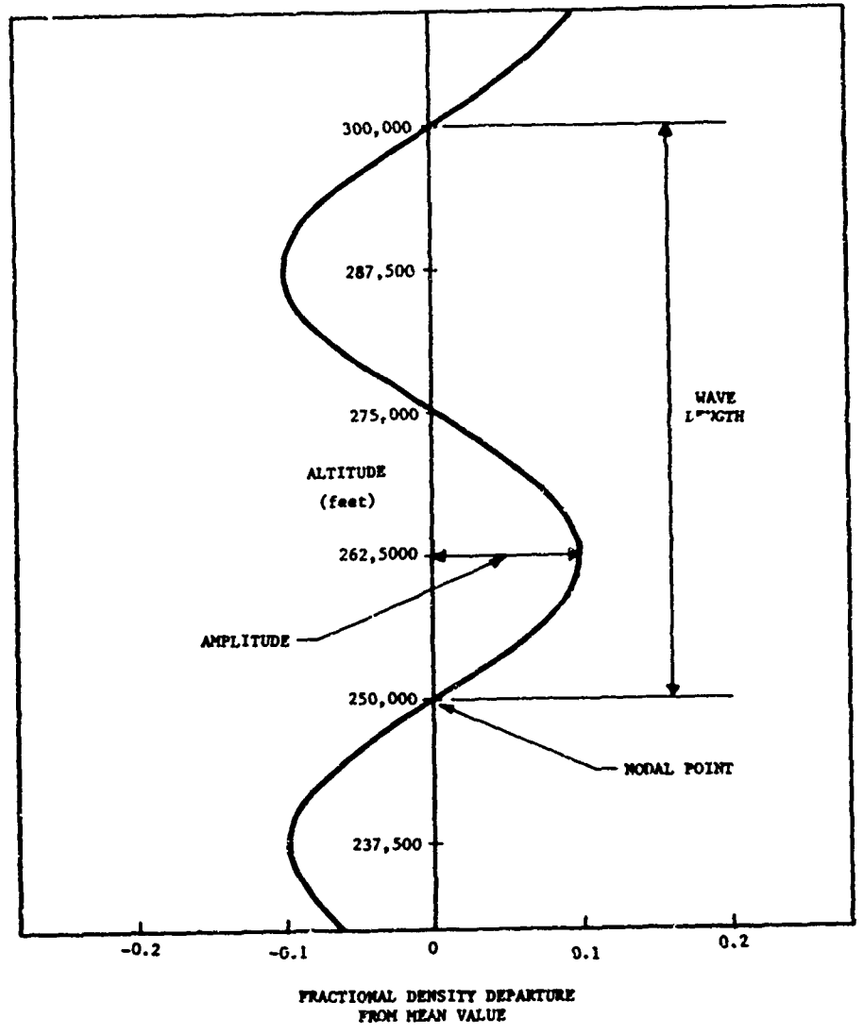


FIGURE 35 FRACTIONAL VARIATION OF DENSITY WITH ALTITUDE

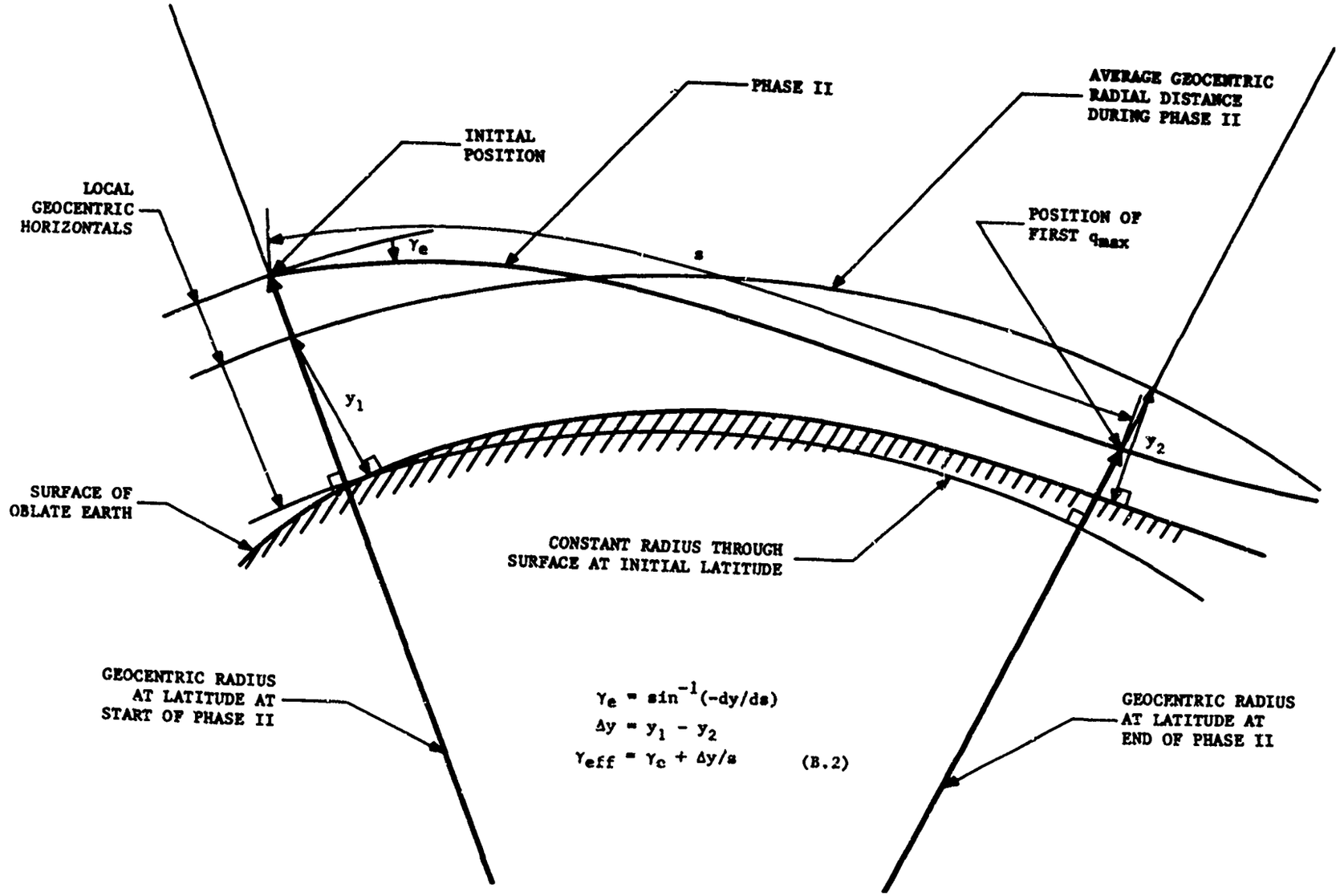


FIGURE 37 DEFINITION OF THE EFFECTIVE INITIAL GLIDE ANGLE

APPENDIX A

The Equilibrium Glide Trajectory

An equilibrium glide trajectory is a coplanar reentry during which the lift is always equal to the net difference between the gravitational and the centrifugal forces on the vehicle. Since drag is not compensated by thrust, the speed is gradually reduced thereby requiring an increase in lift to offset the loss in centrifugal force. Since the angle of attack is fixed, this requirement is met by continually descending to a lower altitude, but a controlled rate so that the lift never becomes too large. As a consequence, the reentry is characterized by the absence of any phugoid or long period oscillations involving the exchange of speed for altitude and vice versa.

For a simple coplanar reentry over a circular non-rotating Earth, the equations of motion are given by

$$m\dot{V} + D - mg\sin\gamma = 0 \tag{A.1}$$

$$mV\dot{\gamma} + L + m\left(\frac{V^2}{r} - g\right)\cos\gamma = 0 \tag{A.2}$$

For an equilibrium glide, all the important effects of interest occur between 100,000 feet and 350,000 feet. Consequently, the values of r and g will vary by less than $\pm 1\%$ from the average values over the above altitude range. Therefore, it is reasonable to make r and g constant.

During an equilibrium glide, it is assumed that the $mV\dot{\gamma}$ term in Eq. (A.2) is negligible compared to the lift. Note that this term cannot always be equal to zero since this would require a constant glide angle.

Finally, the analysis will be restricted to the region for which $\gamma \ll 1$. (For the vehicle used in this investigation, $\gamma < 5^\circ$ until the speed drops to 4300 feet/sec at an altitude of 140,000 feet.) Also, the lift and drag are expressed in the usual manner, in terms of C_L and C_D. Equations (A.1) and (A.2) become

$$\dot{V} + C_D \tfrac{1}{2}\rho V^2 S/m - g\gamma = 0 \tag{A.3}$$

$$C_L \tfrac{1}{2}\rho V^2 S/m = \left(g - \frac{V^2}{r}\right) \tag{A.4}$$

Equation (A.4) is then rewritten to give

$$\rho = \frac{2m}{C_L S r}\,\frac{1 - s^2}{s^2} \tag{A.5}$$

where the speed ratio is

$$s = V/\sqrt{rg} \tag{A.6}$$

Equation (3.7) can be used to find the temperature as a function of speed. The result is

$$T^4 = Ks^{2.15}\,(1-s_i^2)^{\frac{1}{2}} \tag{A.7}$$

where

$$K = \frac{17{,}600\ (\sqrt{rg})^{3.15}}{\epsilon\ \sigma\ \sqrt{R_n}(26{,}000)^{3.15}\sqrt{0.002378\ \frac{C_L S}{2m}\ r}} \tag{A.8}$$

It is straightforward to show that the peak temperature occurs at

$$s^* = \sqrt{2}/3 = 0.8165 \tag{A.9}$$

The use of GM and R_o from equations (4.7) and (4.8), respectively, and using g and r for the average height of 250,000 feet gives the values at the position of the peak temperature as

$$T^* = 3035 \left(\frac{m}{\epsilon^2 R_n C_L S}\right)^{1/8} {}^{\circ}R \tag{A.10}$$

$$V^* = 21{,}050 \text{ feet/sec} \tag{A.11}$$

$$\rho^* = 4.722 \times 10^{-8} \left(\frac{m}{C_L S}\right) \text{ slugs/ft}^3 \tag{A.12}$$

Equations (A.10) to (A.12) show that the peak temperature always occurs at the same speed. Also, the height, y^* (actually the density) is dependent only on the lift coefficient and the wing loading. From Eq. (A.10), the smallest value for T^* is obtained for the maximum value of vertical force, i.e. of $C_L S/m$. Consequently, the smallest T^* is obtained by holding the vehicle at the attitude for $C_{L_{max}}$. The expression for C_L in Eq. (3.3) is a maximum for an angle of attack of 55 degrees, which is, therefore, the optimum angle.

At the optimum angle of 55 degrees, $C_L S/m$ is 0.5417 sq.ft/slug for the subject vehicle. Substitution of this value together with $\epsilon\sqrt{R_n} = 1.0$ into Eqs. (A.10) and (A.12) yields the values in Table A.1.

An equilibrium glide can be realized very closely in practice if the proper initial conditions are achieved at the initial position. Shown on Figure 26 are the values of speed and glide angle which are required at 350,000 feet altitude above a circular, non-rotating Earth. Initially, the speed and glide angle are adjusted to produce a negligible value for $m\dot{V}\dot{\gamma}$ in Eq. (A.2). It is found that the required condition of a negligible $m\dot{V}\dot{\gamma}$ (relative to L) then prevails throughout the flight without any additional control. The values of T^*, V^*, ρ^* are also shown in Table A.1 for the trajectory computed from the exact equations. The value of y^* is also included for this case and may be compared to the value of $y^*(\rho^*)$ which results from Eq. (A.12) when the U.S. Standard Atmosphere, 1962 is the curve of density versus altitude. The same curve was used for the numerically computed trajectory.

The agreement between the two sets of data shows that equations (A.10) to (A.12) are quite accurate. Consequently, the standard peak temperature, T^*, as given by Eq. (A.10) is shown to be independent of (L/D). Also, the smallest value of T^* is achieved for the smallest value of $(m/\epsilon^2 R_n C_L S)$. In addition, because of the 1/8 power, the gains to be realized from even a 50% reduction in this parameter are not large; when T^* equals 3000°F, the reduction in T^* is only 320F° or only a little more than 10%.

TABLE A1

Flight Conditions at the Position of Peak Temperature During an Equilibrium Glide

Value	Approximate Analytical Results[+]	Numerical Integration of Exact Equations[∓]
T^*	$2817^{\circ}F$	$2810^{\circ}F$
V^*	21,050 feet/sec	20,800 feet/sec
ρ^*	8.72×10^{-8} slugs/ft^3	9.27×10^{-8} slugs/ft^3
y^*	245,500 feet***	243,800 feet***

+ equations (A.10) to (A.12)

∓ initial conditions of Fig. 26

*** corresponding to ρ^* for the U.S. Standard Atmosphere, 1962

APPENDIX B

The Effective Initial Glide Angle

The definition of the effective initial glide angle over an oblate Earth is quite straightforward. It simply takes account of the effective change in height of the vehicle which occurs during Phase II of the reentry due to the change in the Earth's radius. This portion of the flight has been chosen since it is during this period that both the peak temperature and the overall down range are determined. As shown in Fig. 37, the change in the Earth's radius may be used to give

$$\sin\Delta\gamma = \frac{\Delta y}{s} \tag{B.1}$$

where Δy is defined in Fig. 37.

For this investigation, both γ and $\Delta\gamma$ are always very small during Phase II. Consequently, the small angle approximation may be used to give

$$\gamma_{eff} = \gamma_e + \frac{\Delta y}{s} \tag{B.2}$$

In practice, Δy is always very easy to find. It is the difference in the height over the Earth's surface when the same change that takes place during Phase II in geocentric latitude also occurs at the average geocentric radius which is used during Phase II. The value of s is the flight distance travelled during this portion of the reentry.

PROGRAM C

Program Listing


```
$JOB  J. H. FINE      F05105 VARIAT     15    5    ALLC'S COMPILE
$EXECUTE        IBJOB
$IBJOB          GO,MAP
$NAME           1.UM05,1=,UM02,
$NAME           [illegible]UM01,1=,UM03,
$NAME           EXIT=......
$IBFTC MCIN     FULIST,PMAP,DD,DECK
C     MAIN PROGRAM CODE NUMBER MCIN                          VERSION 6.0

C      THIS PROGRAM INCLUDES PROVISIONS FOR  -
C         1. A VARIABLE VEHICLE ATTITUDE TO MINIMIZE TEMPERATURE AND
C            ACCELERATION PEAKS AND TO MAXIMIZE CROSS RANGE
C         2. EARTH ROTATION AND OBLATENESS
C         3. EQUIVALENT RANGE COMPUTATIONS AND THEIR OPTIONAL PRINTOUT
C         4. INTERMEDIATE PRINTOUTS AT SPECIFIED POINTS
C         5. CONTINUOUSLY VARIABLE STEP SIZE CONTROL
C      SUMMARY OF SUBROUTINES NOT AVAILABLE FROM LIBRARY
C         A. THE FOLLOWING ROUTINES CONTAIN VITAL PROGRAMMING FEATURES
C            WHICH MUST BE PRESENT SINCE THEY ARE ALWAYS USED  -
C              CALLDA  INITIL  GOPTNS  RUNGEU  OUTPUT  OUTCNT  OUTFRL
C              SNCSGA  SNCSOL  SNCSPS  SNCSPH  SWITCH  AFROND  AFRQKS
C              AFRANS  AF2SAS  AFRAMS  AFR3AS  AFR2NE  AFRINE  AFVOPS
C         B. THE FOLLOWING ROUTINES CONTAIN VITAL PROGRAMMING FEATURES
C            WHICH MUST BE PRESENT IF USED.   THE DUMMY BINARY DECKS WILL
C            HALT COMPUTATION IF THEY ARE USED INSTEAD OF THE ACTUAL
C            ROUTINE WITH THE PROPER PROGRAMMING FEATURES  -
C              CHECKO  CHECKD  CHECK1  CHECK2  CHECKT  SNCSAL  GRAVCF
C              PTTOS   PTTOS1  RADIUS  RANGE   PITCH   PITCHI  PITCH[illegible]
C         C. THE FOLLOWING ROUTINES ALLOW CONTROL TO PROCEED THROUGH THEM
C            WHEN THEY ARE USED IF NO ACTUAL VITAL PROGRAMMING FEATURES
C            NECESSARY  -
C              CONSTS  DE[illegible]  DENVPI  GRAVC1  GRAVCP  GRAVCF  PAUSE
C              RANGPI  RANGER  FINAL   PRGLST  EQNLST  TPELST  ENDLST
C      SUMMARY OF SENSE SWITCH CONTROL SETTINGS
C         A. FOR PROGRAM CONTROL
C            (1) AT 944 TO TERMINATE CASE WITH OR WITHOUT FIVE MORE STEPS
C            (2) AT 946 TO LIMIT CASE TERMINATION TO ONE MORE STEP
C            (3) AT 962 TO TERMINATE COMPUTATION AT END OF CURRENT CASE
C            (4) AT 980 AND 976 TO PRINT OPERATOR CASE MESSAGE WHEN
C                               KNTMAX IS ZERO AND POSITIVE RESPECTIVELY,
C                AT 974 TO STOP THE PRINTING OF OPERATOR CASE MESSAGE
C                               WHEN KNTMAX IS NEGATIVE
C         B. FOR SPECIAL DEBUG OUTPUTS, PROGRAM GOES TO GIVEN STATEMENT
C            WHERE SUBROUTINE SWITCH DETERMINES WHETHER OR NOT THE NEXT
C            STATEMENT IS EXECUTED OR SKIPPED
C              1301      1901      2901      360.
C              4001      4401      6901      6201
C              8001      8101      8901      9101      9901
```



```
C      COMMON STATEMENTS FOR GLOBAL VARIABLES
       COMMON /CNSS/     NNPPTS,NSTOP
       COMMON /CONTRL/   SNSSWT,TNCNTS,GOPTNS,NOCASE,KNTMAX,THE,SNATOT,
      1                  FNSHTS,TPOCTS,THPSTS,DEGATS,ADTSTS,DTSNTS
       COMMON /CFORMT/   F05(25),F09(25)
       COMMON /COTCTL/   IOTCTL(15)/COTFIL/IOTFIL(50)/COTF2L/IOTF2L(50)
       COMMON /CVAR/     V20,V0,R0,GAMMA0,PSI0,DELTA0,THETA0,TIME0,BETA0,
      1                  ENDURO,TEMP0,S0
       COMMON /CIVAR/    VINIT0,V2NIT0,GDINIT,PDINIT,DDINIT,TDINIT
       COMMON /CTVAR/    V2TEMP,RTEMP0,GTEMP0,PTEMP0,DTEMP0,TTEMP0,TITEMP,
      1                  BTEMP0,ETEMP0,STEMP0
       COMMON /CFVAR/    FV20,FV0,FR0,FT0,FTP0,FG0,FP0,FD0,FT0,FTI0,F00,
      1                  FE0,FV2P15,ATQDTY
       COMMON /CAVAR/    AV20,AR0,AG0,AP0,AD0,AT0,ATI0,A00,AE0,ATP0,DS0
       COMMON /COVAR/    DVT(4),DP(4),DG(4),DP(4),DD(4),DT(4),DTI(4),
      1                  DO(4),DE(4),HTP(6,4)
       COMMON /CSCGPD/   PDOLD,SINE,COSG,PPOLD,SINP,COSP,PDOLD,SIND,COSD
       COMMON /CSCAP/    ALPHAD,ALPOLD,SINALP,COSALP,PHIDEG,PHIOLD,SINPHI,
      1                  COSPHI
       COMMON /CINTS/    M,N,NTP,IGPDYD,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
      1                  ICNTRL,NMPTEV,IDSTOP,KOUNTF,KOUNTL,NSTP,NUSL,
      2                  NSTPTT,NUSLTT,IT,ITSAVE
       COMMON /CPIES/    PIE0,PIE020,PIET20,P[illegible],[illegible],P[illegible],[illegible],
      1                  PIE02L,PIE02H,PIEL,PIEH,PIET2L,PIET2H

C      TYPE STATEMENTS FOR GLOBAL VARIABLES
       DOUBLE PRECISION V20,V0,R0,GAMMA0,PSI0,DELTA0,THETA0,TIME0,BETA0
       DOUBLE PRECISION ENDURO,TEMP0(6),S0
       DOUBLE PRECISION VINIT0,V2NIT0,GDINIT,PDINIT,DDINIT,TDINIT
       DOUBLE PRECISION V2TEMP,RTEMP0,GTEMP0,PTEMP0,DTEMP0,TTEMP0,TITEMP
       DOUBLE PRECISION BTEMP0,ETEMP0,STEMP0
       DOUBLE PRECISION FV20,FV0,FR0,FT0,FTP0,FG0,FP0,FD0,FT0,FTI0,F00
       DOUBLE PRECISION FE0
       DOUBLE PRECISION AV20,AR0,AG0,AP0,AD0,AT0,ATI0,A00,AE0,ATP0(6),DS0
       DOUBLE PRECISION PIE0,PIE020,PIET20,[illegible],[illegible]
       REAL             TEMPS(2,6),ATPS(2,6)

C      EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
       EQUIVALENCE      (V2,V20),(V,V0),(R,R0),(GAMMA,GAMMA0),(PSI,PSI0)
       EQUIVALENCE      (DELTA,DELTA0),(THETA,THETA0),(TIME,TIME0)
       EQUIVALENCE      (BETA,BETA0),(ENDURE,ENDURO),(TEMPS,TEMP0),(S,S0)
       EQUIVALENCE      (V2TEMP,V2TEMP),(RTEMP,RTEMP0),(GTEMP,GTEMP0)
       EQUIVALENCE      (PTEMP,PTEMP0),(DTEMP,DTEMP0),(TTEMP,TTEMP0)
       EQUIVALENCE      (TITEMP,TITEMP),(BTEMP,BTEMP0),(ETEMP,ETEMP0)
       EQUIVALENCE      (STEMP,STEMP0)
       EQUIVALENCE      (FV2,FV20),(FV,FV0),(FR,FR0),(FT,FT0),(FTP,FTP0)
       EQUIVALENCE      (FG,FG0),(FP,FP0),(FD,FD0),(FT,FT0),(FTI,FTI0)
       EQUIVALENCE      (FO,F00),(FE,FE0)
       EQUIVALENCE      (AV2,AV20),(AR,AR0),(AG,AG0),(AP,AP0),(AD,AD0)
       EQUIVALENCE      (AT,AT0),(ATI,ATI0),(AO,A00),(AE,AE0),(ATPS,ATP0)
       EQUIVALENCE      (DS,DS0)
       EQUIVALENCE      (PIE,PIE0),(PIE02,PIE020),(PIET2,PIET20)
```



```
C      SUMMARY OF VALUES WHICH MAY BE CHANGED UNDER CONTROL OF GOPTNS

C      GOPTNS  STATEMENT FORMAT LIST OF VALUES

C         1       123      24    VHIMOL  VHINF   VHIML   DVMIN
C                                PHMNOL  PHLMNF  PHLMNL  DMACHN
C         2       126      26    YTPOPL  ENGYOL  ENGYA   ENGYL
C                                DEGYOL  DEGCOL  ENDCOM
C         4       128      28    PYPLUT  YMAXF   YMAXOL
C                                VMINOL  VMINF   VMINL   DYMIN

C        10       132      32    APLOUT  AMAXF   AMAXOL  AMINOL  AMINF
C                                DAMIN   TTAZOL  ACCLAS(9)
C                                ENDR12  ENDR22  ENDR3   ENDR42  ENDR3
C                                ENDR6   ENDRT   EDMXOL  ENDMXF  ENDMPY
C        20       134      34    TIMAXF  TIMAXL  TIMXOL  DTIMIN
C                                QMAXF   QMAXOL  DQMIN (VALUES IN DEGREES)
C                                SQMAXF  SQMAXL  SQMXOL  SNTEND  DSMAXS
C                                IOS(3)  AOTEST  AGTXOL  AGSPOL  ADGCOM
C        40       136      36    PPLOUT  PMAXF PMAXOL  PMINOL PMINF  DPMIN
C                                TPLOUT  TMAXF TMAXOL  TMINOL TMINF  DTMIN
C                                NOTE -  FOR PHI(7)  =  -1, 0  VALUES ARE
C                                        TO BE GIVEN IN DEGREES.
C                                     -  FOR PHI(7)  =  +1  VALUES ARE TO
C                                        BE GIVEN IN MULTIPLES OF  PIE .

C       100       142      42    ATMAOS(6) CNDS(6)    E(6)       TEMPDS(6)
C                                TTEOBL    TEMPOS(3)  TPMDS(9)   TPMCAS(3)
C       200       146      46    MNDERO    DOTVIN     S/ERND
C                                DDAVE2    DRPPR.     SDALTN
C                         CONSTS ENDZN1  BETA1N  BETA2N  FYSPLT  TEST
C       400       149     CONSTS STPTIN  VAPNCE  PPCMIN  PPCMAX  SINCO
C                                AMPFIX  AMPSIN  WAVELN  FYMODE  INCPHZ
C                         CONSTS AMPLDE(INEPHZ)
C                         CONSTS FYREF(INEPHZ)

C      1033       103      13    AGICIN  ATICIN  TPICIN  SNGSWT  TRNMAX
C                                PNSHTS  FTPOPL  FYPLUT  TPOCTS  THPSTS
C                                PPLOUT  TPLOUT  DEGATS  ADTSTS  ACELTS
C                                APLOUT  ACEXTS  DTSNTS  TEMPTS  TPEXTS
C                                THEXTS  TCEXTS
C      2003       155     PITCHI GROPAZ  WNS     COEFRO  CP      CPGRAG
C                                DALPHS  DALPTS  ALPRAT  PHIRAT  PHIRTE
C                                BXOL    CONTOL  PI2SOL  ROLD    ALPHAI
C                                MODI    PRTOLI  PRTOL2  PRGCOL  CSCPOL
C                                PTEADJ  FYMIN   FYMAX   FYMADF  FYMIN
C                                FYMAX   FYMADF  GAINOL  FGOMIN  FGOMAX
C                                FEGAND  ALPOHL  ALPOH2  ALPOHI  ALPODS
C                                RATMPY  TIMMIN  MNGAD   NGMAX   MULTMD
C                                RIMPRC  RAIPRC  ALPSUM  PHISUM  GAMMAL
C                                GNAMPH  FYTMAX  PYTMAX  PROTNX
C                                NOTE -  EXCEPT FOR GROPAZ, THESE VALUES
C                                ARE CHANGED ONLY WITHIN SUBROUTINE PITCH
C      4003       155     PITCHI ALPHAF  MODE    ALPHAS  MODS    PHISUP
C                                ALPHAI  MODI    ICPSOL
```



```
C      COMMON STATEMENTS FOR GOPTNS VARIABLES
       COMMON /C001/     VMIN,VMIN2,VMINOL,VMINF,VMINL,DVMIN,
      1                  PMACHN,PMACHP,PMNOUT,PHLMNN,PHLMN2,PHMNOL,PHLMNF,
      2                  PHLMNL,DMACHN
       COMMON /C002/     YTPOPL,ENERGY,TENRGY,ENGOUT,ENGYOL,ENGYF,ENGYA,
      1                  ENGYL,DENRGY,DEGYOL,DEGYCM,DEGCOL,ENDCOM
       COMMON /C004/     FYPLUT,YMAX,YMAXF,YMAXOL,YMIN,YMINOL,YMINF,YMINL,
      1                  DYMIN
       COMMON /C010/     ACELTS,APLOUT,ACEXTS,ACCEL,ACCEL2,AMAX,AMAX2,
      1                  AMAXF,AMAXOL,AMIN,AMIN2,AMINOL,AMINF,DAMIN,
      2                  TESTAZ,TTAZOL,ACCLAS(9),ANGLD2,ALDLDE,ACC2OL,
      3                  ACC2HL,ACC2MX,ACC2MF,ENDOUT,ENDR12,ENDR22,ENDR3,
      4                  ENDR42,ENDR3,ENDR6,ENDRT,ENDMAX,EDMXOL,ENDMXF,
      5                  ENDMPY
       COMMON /C020/     TIMAX,TIMAXF,TIMAXL,TIMXOL,DTIMIN,QMAX,QMAXF,
      1                  QMAXOL,DQMIN,SNGOUT,SQMAX,SQMAXF,SQMAXL,SQMXOL,
      2                  SNTEND,DSSAVE,DSMAXS,DSMAX,DSMIN,DSMIND,DSMINF,
      3                  DSMINT,DSMND,AOTEST,AGNCGT,AGTEST,AGTXOL,AGSTNP,
      4                  AGSPOL,AGAXGT,ADGCOM
       COMMON /C040/     PPLOUT,PMAX,PMAXF,PMAXOL,PMIN,PMINOL,PMINF,DPMIN,
      1                  TPLOUT,TMAX,TMAXF,TMAXOL,TMIN,TMINOL,TMINF,DTMIN
       COMMON /C0100/    TEMPTS,TPEXTS,THEXTS,TCEXTS,TPMDRQ,TAMDRQ,TPEONT,
      1                  TPMAX(6),ATPHTT(6),ATPLTT(6),P1(6),P2(6),P0(6),
      2                  P1COST,P2CMST,P0CMST,ATMAOS(6),CNDS(6),E(6),
      3                  TEMPIN,TESTEQ,TTEOBL,TEMPOS(3),TPMDS(9),
      4                  TPMCAS(3),TAMDRL,TALMDL,TAMDOM,TAMDOL,TAMDRE,
      5                  TAMCAE,TAMDRF,TAMFAP
       COMMON /C0200/    BNDZN1,BETA1
C      DOUBLE PRECISION STATEMENTS FOR GOPTNS VARIABLES
       DOUBLE PRECISION SNMAX,SQMAXF,SQMAXL,SQMXOL,SNTEND
       DOUBLE PRECISION TEMPIN(6)

C      EQUIVALENCE STATEMENTS FOR GOPTNS VARIABLES
       EQUIVALENCE      (SNTEN,SNTEND),(ACZMF,ACC2MF),(TMDRF,TAMDRF)
```

2


```
C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /COSSET/  AMBINC,ATALIC,ATPHIC,ATPSIC,TEPINC,DLTEST,DSMIN,
     1                 DMAX1,DMAX2,DMARC,
      COMMON /COCTAL/  DMACO,DRTS,CR3774,C0777
      COMMON /CPTTOS/  TOENTY,PRESS,TAML,TATM,DENSTY,SNDSPD,SDSPD2,UNBL,
     1                 VEAN,IPTOS1,IPTOS2,IPTOS3,IPTOS4
      COMMON /CPAMUS/  SHAPE,DSHAPE,PEQUE,PEQU20,PYD1,PYD2,PYD3,PYD4,
     1                 PYS01,PYS02,PYS03,PYS04,PYS05,PEARTH,REARTH,
     2                 THETAZ,RZERO,RZ20
      COMMON /CAMODE/  GITPLE,DOMNP,COSGSR,PSIND,PSIOGA
      COMMON /CROTA/   RDOT,RDOT,WD,W2,TW,TWZ,ROTR,ROTP,ROTV,SORLTR
      COMMON /CGRAV/   DGRAV,GRAVG,GRAVP,GRAVV,GNPGRD,GRAVRD,GRAVO,
     1                 GRAVT,SINDE,GND,GDZVJ0,GDZVJ2,GDAVR2,GREFRZ,
     2                 TERPR2
      COMMON /CPITCH/  ATTACK,ATAKNE,ATKALP,ATKPHI,BANK,CASMAX,ROLL,
     1                 COSRH,COSR2R,CLSDPR,MOD,DPAS,HIPT,HIPTSP,HIPTCP,
     2                 PSITGT,PSIRUC,CSCPOR,ALPRAT,PHIRAT,PHIRTE,STASPR

C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION REQUO,REQU20,PYD1,PYD2,PYD3,PYD4,RZEROD,RZ20
      DOUBLE PRECISION W3,OMEGRD,GRAVRD,GND,GDZVJ0,GDZVJ2,GDAVR2

C     EQUIVALENCE STATEMENTS FOR SUBROUTINE VARIABLES
      EQUIVALENCE      (IDLTST,DLTEST)
      EQUIVALENCE      (REQU,REQUO),(REQU2,REQU20),(PYS01,PYD1)
      EQUIVALENCE      (PYS02,PYD2),(PYS03,PYD3),(PYS04,PYD4)
      EQUIVALENCE      (RZERO,RZEROD),(RZ2,RZ20)
      EQUIVALENCE      (W,W3),(OMEGPR,OMEGRD),(GRAVR,GRAVRD),(GN,GND)
      EQUIVALENCE      (GDZVJ0,GDZVJ0),(GDZVJ2,GDZVJ2),(GDAVR2,GDAVR2)

C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE      (AA,OUT)

C     INITIALIZATION BEFORE EACH CASE SENSE SWITCH CONTROL
 100  IF(   SWSSWT    ) 1001,1003,1001
 1001 IF( SWITCH(100) ) 1002,1003,1002
 1002 AA = OUT
 1003 CONTINUE

C     INITIALIZATION BEFORE EACH CASE
 101  CALL INITIL
      GO TO 400

C     EVALUATION OF THE INCREMENTS SENSE SWITCH CONTROL
 200  IF(   SWSSWT    ) 2001,2003,2001
 2001 IF( SWITCH(200) ) 2002,2003,2002
 2002 AA = OUT
 2003 CONTINUE
C     EVALUATION OF THE INCREMENTS

 201  CALL RUNGKU
```



```
C     OUTPUT CHECK CONTROL SENSE SWITCH CONTROL
 400  IF(   SWSSWT    ) 4001,4003,4001
 4001 IF( SWITCH(400) ) 4002,4003,4002
 4002 AA = OUT
 4003 CONTINUE

C     PRELIMINARY CALCULATIONS FOR OUTPUT CHECK CONTROL
 401  V2TEMP =  V20     +   AV20
      RTEMPD =  RP      +   ARD
      GTEMPD =  GAMMAD  +   AGD
      PTEMPD =  PSID    +   APD
      DTEMPD =  DELTAD  +   ADD
      TTEMPD =  THETAD  +   ATD
      TITEMP =  TIMED   +   ATID
      BTEMPD =  BETAD   +   ABD
      ETEMPD =  ENRGD   +   AED
      STEMPD =  SD      +   ASD
 402  PV20   =  V2TEMP
      PV     =  SQRT(PV2)
      PGD    =  GTEMPD
      PPD    =  PTEMPD
      PDD    =  DTEMPD
      PTD    =  TTEMPD
      PTID   =  TITEMP
 403  IF(   DMOD(DSHAPE)   ) 404, 405, 404
 404  PRD    =  RTEMPD
      GO TO 406
 405  PYDR   =  RTEMPD  -   RZEROD
      PY     =  PYD
      PR     =  PY      +   RZERO
 406  IEPOYD =  - 100
 407  CALL RUNGKU

 411  IF(   DOBL(TOENTY)   ) 412, 440, 412
 412  IF(   DOBLE( HTP )   ) 413, 414, 413
 413  TAMBEQ =  PDCRST  *   RTAMTY  *  PV2P15  *  PV
 414  PMACH2 =  PV2  /  SDSPD2

C     CHECKS FOR ATTITUDE CONTROL
 420  IF(   DOBL( ROLL )   ) 4201,440, 4201
 4201 CALL CHECKR
 4202 IF(       IEPOYD     ) 440, 609, 000

C     OUTPUT CHECK CONTROL SENSE SWITCH CONTROL
 440  IF(   SWSSWT    ) 4401,4403,4401
 4401 IF( SWITCH(440) ) 4402,4403,4402
 4402 AA = OUT
 4403 CONTINUE
```



```
C     CHECKS ON THE TEMPORARY VALUES BEFORE POINT EVALUATION

C     CHECKS ON THE ENERGY, MACH NUMBER, VELOCITY, AND HEIGHT
 500  IF(   DOBL(ELSWTS)   ) 5001,510, 5001
 5001 CALL CHECKO
 5002 IF(       IEPOYD     ) 510, 600, 000

C     CHECKS ON THE LONGITUDE AND AZIMUTH
C     CHECKS ON THE LATITUDE AND  FLIGHT PATH ANGLE
 510  IF(   DOBL(TPOSTS)   ) 5101,520, 5101
 5101 CALL CHECK1
 5102 IF(       IEPOYD     ) 520, 600, 000

C     CHECKS ON THE ACCELERATION
C     CHECKS ON THE TIME, RADIAL ANGLE, AND PATH LENGTH
 520  IF(   DOBL(ADTSTS)   ) 5201,530, 5201
 5201 CALL CHECK2
 5202 IF(       IEPOYD     ) 530, 600, 000

C     CHECKS ON THE TEMPERATURES
 530  IF(   DOBL(TEMPTS)   ) 5301,541, 5301
 5301 CALL CHECKT
 5302 IF(       IEPOYD     ) 541, 600, 000

C     FINAL CHECK CONTROL FOR THE TEMPORARY VALUES
 541  IF(       INTLST     ) 542, 543, 542
 542  INTLST =  0
      ICNTRL =  702
      JFAIL  =  JFAIL   +   1
      INTCTL(JFAIL)  =  ICNTRL
 543  IF( SWSSWT ) 544, 547, 544
 544  IF( SWITCH(1)   ) 545, 547, 545
 545  ICNTRL =  799
      IF( SWITCH(2)   ) 901, 546, 901
 546  KOUNTP =  KOUNTP  +  1
      IF( KOUNTP  -  4 ) 902, 902, 901
 547  IF( -DMAX1 +  ABS(DTEMP)   ) 548, 548, 597
 548  IF(       DLTEST         ) 549, 613, 549
 549  IF( -DMARC +  ABS(DTEMP)   ) 612, 611, 611

C     TRANSFER CONTROL FOR CHECKS ON THE TEMPORARY VALUES
 597  NOPTEV =  597
      IF( DSMIND - DS ) 607, 611, 611
```



```
C     POINT CHECK CONTROL IMPLEMENTATION

C     POINT CHECK CONTROL IMPLEMENTATION SENSE SWITCH CONTROL
 600  IF(   SWSSWT    ) 6001,6003,6001
 6001 IF( SWITCH(600) ) 6002,6003,6002
 6002 AA = OUT
 6003 CONTINUE

C     NO POINT EVALUATION - OUTPUT CHECK CONTROL NOT PASSED
 601  NUSL   =  NUSL   +  1
      IOTFIL(NUSL)      =   NOPTEV
      IOTF2L(NUSL)      =   KOUNT
      DG     =  DG   *   DSRED
      DP     =  DP   *   DSRED
      DD     =  DD   *   DSRED
      DV2    =  DV2  *   DSRED
      DR     =  DR   *   DSRED
      DT     =  DT   *   DSRED
      DTI    =  DTI  *   DSRED
      DA     =  DA   *   DSRED
      DE     =  DE   *   DSRED
      PYP    =  PYTEMP
      IF(  HTP    ) 602, 200, 602
 602  DO 603  IT = 1, HTP
 603  PTP(IT,1)  =  DTP(IT,1)   *   DSRED
      GO TO 200

 607  IGNORE =  (1-ISIGN(1,IGNORE))*(1-ISIGN(1,IDLTST)) / 4  +  IGNORE
      DLTEST =  1.0
      DMARC  =  AMIN1( ABS(DELTA), DMAX1 )
      DSRED  =  DSMIND / DS
      DS     =  DSMIND
      DSMAX  =  DSMIND
      IEPOYD =  0
      GO TO 600

C     POINT EVALUATION INTERVAL SIZE CONTROL
 611  DSMAX  =  DSMIND
      IF( IGNORE ) 617, 620, 615
 612  DLTEST =  0.0
      DMARC  =  DMAX1
 613  DSMAX  =  DSMAXS
 614  IF( IGNORE ) 617, 620, 615
 615  IF(DSMIND  -    DS   ) 616, 620, 620
 616  DS     =  0.5 * DS
      GO TO 620
 617  IF( MOD( NSTP, 100 ) - 40 ) 620, 620, 618
 618  DS     =  2.0 * DS
      NSTP   =  NSTP    +   90
      GO TO 620
```

3


```
C     POINT CHECK CONTROL IMPLEMENTATION SENSE SWITCH CONTROL
 620  IF(   SNSSWT    ) 6201,6203,6201
 6201 IF( SWITCH(620) ) 6202,6203,6202
 6202 AA = OUT
 6203 CONTINUE

C     POINT EVALUATION - OUTPUT TEST ON JFAIL AT END OF SEQUENCE
 621  V20    =  V2TEMP
      RD     =  RTEMPO
      GAMMAO =  GTEMPO
      PSIO   =  PTEMPO
      DELTAO =  DTEMPO
      THETAO =  TTEMPO
      TIMEO  =  TITEMP
      BETAO  =  BTEMPO
      ENOUAD =  ETEMPO
      SO     =  STEMPO
      FVTEMP =  FV
      KOUNT  =  IABS(I) +   KOUNT
      BIASPR =  BIASPR  +  ANGINC
      AMOLD2 =  SIGN(   AMAX1( AMOLD2, ACCEL2 ),   TESTA2   )
      ALOLD2 =  AMAX1(  SIGN( 1.0E6, TESTA2 ),  AMIN1(ALOLDE,ACCEL2)  )

      IF(   BOOL( NTP  )    ) 622, 626, 622
 622  IF(         NTP       ) 623, 625, 623
 623  DO  624  IT = 1, NTP
 624  TEMPO(IT)    =  TEMPO(IT)    +    ATPO(IT)
 625  TANOL  =  SIGN(   AMAX1( TANNOL, TANDEO ),   TESTEO   )
      TALNOL =  AMAX1(  SIGN( 1.0E20, TESTEO ), AMIN1(TALNOL,TANDEO)  )

 626  IF( ATEALP +  ATAPHI ) 627, 631, 627
 627  CALL PITCHA

 631  IF( IGNORE ) 632, 635, 632
 632  NSTP   =  NSTP  +  1
      IF( 45 - NSTP + NSTP/100000 - (IGNORE+1)/2+NUSL ) 633, 638, 638
 633  ICNTRL =  NOPTEV
      JFAIL  =  JFAIL   +   1
      IOTCTL(JFAIL)  =  ICNTRL
      WRITE(6,84)  NUSL,NSTP,(IOTP1L(K),IOTP2L(K),K=1,NUSL)
      KOUNTF =  KOUNTF    +   1
      ILAST  =  KOUNTF / 15
      IF( IDSTOP ) 634, 810, 810
 634  IDSTOP =  0
      GO TO 810
 635  DS     =  1.0
      IF( IDSTOP ) 636, 638, 637
 636  IDSTOP =  0
 637  NSTP   =  NSTP  +  100
      IF( NSTP - NSTP/100000 + NUSL - 45 ) 638, 633, 633
 638  IF( JFAIL  ) 810, 200, 810
```



```
C     OUTPUT CONTROL SENSE SWITCH CONTROL
 800  IF(   SNSSWT    ) 8001,8003,8001
 8001 IF( SWITCH(800) ) 8002,8003,8002
 8002 AA = OUT
 8003 CONTINUE

C     OUTPUT CONTROL
 801  IGPOYO =  - IGPOYO
      JFAIL  =  JFAIL   +   1
      IOTCTL(JFAIL)  =  ICNTRL
      IF( ILAST  ) 802, 803, 620
 802  ILAST  =  1
      FV     =  SQRT(V2)
      FV     =  FVTEMP
      GO TO 210
 803  DS     =  0.0
      KOUNTL =  KOUNT  +  1
      IF( 16 -  JFAIL ) 804, 804, 509
 804  WRITE(6,84)  NUSL,NSTP,((IOTP1L(K),IOTP2L(K),K=1,NUSL)
      WRITE(6,84)  JFAIL,((IOTCTL(K),K=1,JFAIL)
  84  FORMAT(  1H0, I5, 1916 / ( 1X, I5, 1916 )    )
      FVNEW  =  FV
      AC2NEW =  ACCEL2
      TANEW  =  TANDEO
      FV     =  SQRT(V2)
      FV     =  FVTEMP
      ACCEL2 =  ABS( AMOLD2 )
      TANDEO =  ABS( TANNOL )
      ILAST  =  - 1
      GO TO 810
 805  FV     =  SQRT(V2TEMP)
      FV     =  FVNEW
      ACCEL2 =  AC2NEW
      TANDEO =  TANEW
      ILAST  =  1
      GO TO 620


C     FINAL OUTPUT CONTROL
 901  JFAIL  =  JFAIL   +   1
 C3?  IOTCTL(JFAIL)  =  ICNTRL
      ILAST  =  1
      GO TO 620
```



```
C     OUTPUT OF INTERMEDIATE VALUES SENSE SWITCH CONTROL
 810  IF(   SNSSWT    ) 8101,8103,8101
 8101 IF( SWITCH(810) ) 8102,8103,8102
 8102 AA = OUT
 8103 CONTINUE

C     OUTPUT OF INTERMEDIATE VALUES
 811  CALL OUTPUT

C     INITIALIZATION OF INTERMEDIATE CONTROL VALUES
 891  IF( ILAST  ) 805, 892, 910
 892  DS     =  1.0
      IGNORE =  0
      JFAIL  =  0
      NSTP   =  0
      NUSL   =  0
      IF( N  -  NSTOP ) 200, 910, 910


C     OUTPUT OF FINAL AND STORED VALUES SENSE SWITCH CONTROL
 910  IF(   SNSSWT    ) 9101,9103,9101
 9101 IF( SWITCH(910) ) 9102,9103,9102
 9102 AA = OUT
 9103 CONTINUE

C     OUTPUT OF FINAL AND STORED VALUES
 911  CALL OUTFNL
 912  IF( N  -  NSTOP ) 100, 913, 914
 913  IF( SWITCH(913) ) 100, 914, 100
 914  N      =  0
      IF( ABS(GITVLE).LT.2.0.OR.ILAST.GT.0.OR.NSTOP.LE.10 ) GO TO 100
      GITVLE =  SIGN(  ABS( GITVLE )  -  2.0,  GITVLE  )
      VO     =  DSQRT( V20 )
 915  WRITE(6,95)  MDCASE,GITVLE,VO,RD
  95  FORMAT(  17H1      CASE NUMBER, I8, 5X, 11H(CONTINUED), F20.0     /
     1          29H0  CONTINUED VELOCITY  =, 5PD22.10, 9H  FT/SEC., 5X,
     2          19HCONTINUED GEOCENTRIC RADIUS  =, 8PD24.9,  7H  FEET   /
     3          1H0, 22X, 9HGLIDE, 37X, 16HA N G L E S  O F             /
     4          54H  VELOCITY  ALTITUDE   ANGLE   AZIMUTH LATITUDE LONGIT
     5          51HUDE   TIME   ATTACK   ROLL  NO. ACCEL  TEMP(S) CNT    )
      IF(   BOOL(GITVLE)   ) 916, 916, 916
 916  IF(        GITVLE    ) 917, 917, 918
 917  WRITE(6,97)
  97  FORMAT(  1H0, 104X, 11H NSTP NUS J,  6( 4H IOT )                 /
     1          23H  (FT/SEC)    (FEET)     , 4( 5H(DEG), 4X ), 9H(SEC)
     2          15H(DEG)  (DEG)        (G*S)  (DEG-R)                  )
      GO TO 919
 918  WRITE(6,98)
  98  FORMAT(  1H0, 104X, 29H  DOWNR   CROSSR    PSIRG                  /
     1          23H  (FT/SEC)    (FEET)     , 4( 5H(DEG), 4X ), 9H(SEC)
     2          15H(DEG)  (DEG)        (G*S)  (DEG-R)    , 3( 4X5H(DEG) )  )
 919  CALL OUTCNT
      GO TO 200

      END
```

4


```
$IBFTC CCLL     FULIST,MAP,DECK
      BLOCK DATA

C     SUBROUTINE CALLDA CODE NUMBER CCLLDA                  VERSION 6.0

C     THIS PROGRAM WILL INITIALIZE THOSE DATA VALUES WHICH ARE IN
C     LABELED COMMON.

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CNSS/    NWPPTS,NSTOP
      COMMON /CONTRL/  SNSSWT,TMCNTS,GOPTNS,NOCASE,DDUMB(3)
      COMMON /CFORMT/  FOS(25),FO9(25)
      COMMON /CFVAR/   FDUMP(2),FVD,FDUMM(6),FGO,FPO,FBO,FDUMB(18)
      COMMON /CSCBPD/  FGOLD,SING,COSG,FPOLD,SINP,COSP,FOOLD,SINO,COSO
      COMMON /CSCAP/   ALPHAD,ALPOLD,SINALP,COSALP,PHIDEG,PHIOLD,SINPHI,
     1                 COSPHI
      COMMON /CPIES/   PIEO,PIE02D,PIET2D,RDODGD,DGORDD,RDODGP,DGORDP,
     1                 PIEO2L,PIEO2H,PIEL,PIEH,PIET2L,PIET2H
      COMMON /CINTS/   N,M,NTP,IDUMM(4),IDSTOP,IDUMB(8)

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION FVD,FGO,FPO,FBO
      DOUBLE PRECISION PIEO,PIE02D,PIET2D,RGODGD,DGORDD

C     DATA STATEMENTS FOR GLOBAL VARIABLES
      DATA   NWPPTS,NSTOP/512,5H/
      DATA   SNSSWT,GOPTNS,NOCASE/0.0,0.0,-1/
      DATA   N,M,NTP,IDSTOP/0,0,6,0/
      DATA   FOS(1)/113H(I6, A4, I7, A3, F6.,A6, I6, A5, I3, A5, I4, A5,
     1 I9,A4,I3, A5,I4, A5,I3,I2, A6,I3, A4,I4,I6,I4,I2,4I4/(100X,5I4))/
      DATA   FO9(1)/113H(I6, A4, I7, A3, F6.,A6, I6, A5, I3, A5, I4, A5,
     1 I9,A4,I3, A5,I4, A5,I3,I2, A4,I5, A6,I4,  I4, A5,I3, A6,I4, A5 )/
      DATA   FVD,FGO,FPO,FBO/400,900/
      DATA   FGOLD,SING,COSG,FPOLD,SINP,COSP,FOOLD,SINO,COSO/200.0,1.0,
     1       200.0,1.0,200.0,1.0/
      DATA    ALPHAD,ALPOLD,SINALP,COSALP,PHIDEG,PHIOLD,SINPHI,COSPHI/
     1       900.0,1.0,900.0,1.0/
      DATA   PIEO,PIE02D,PIET2D,RDODGD,DGORDD,RDODGP,DGORDP,PIEO2L,
     1       PIEO2H,PIEL,PIEH,PIET2L,PIET2H/3.14159265358979323846,
     2       1.57079632679489661923,6.28318530717958647692,
     3       0.017453292519943295769,57.295779513082320876,
     4       0.017453292927,57.2957794,0201622077324,0201622077326,
     5       0202622077324,0202622077326,0203622077324,0203622077326/
```



```
C     SUMMARY OF VALUES WHICH MAY BE CHANGED UNDER CONTROL OF GOPTNS

C     GOPTNS  STATEMENT FORMAT LIST OF VALUES

C        1       123      24   VMINOL  VMINF   VMINL   DVMIN
C                              FMNMOL  FMLMNF  FMLMNL  DMACHN
C        2       126      26   FSPOPL  ENGYOL  ENGYR   ENGYL
C                              DEG OL  DEGCOL  ENGCON
C        4       123      28   FYFLWT  YMAXF   YMAXOL
C                              YMINOL  YMINF   YMINL   DYMIN

C       10       132      32   AFLOWT  AMAXF   AMAXOL  AMINOL  AMINF
C                              DAMIN   TTAZOL  ACCLBS(9)
C                              ENOR12  ENOR22  ENOR3   ENOR42  ENOR5
C                              ENOR6   ENOR7   EDMXOL  ENDMXF  ENDMXY
C       20       134      34   TIMAXF  TIMAXL  TIMXOL  DTIMIN
C                              BMAXF   BMAXOL  DBMIN (VALUES IN DEGREES)
C                              SDMAXF  SDMAXL  SDMXOL  INTERD  DSMAXS
C                              IDS(5)  ADTEST  AGTSOL  AGSPOL  ADGCON
C       40       136      36   PFLOWT  PMAXF PMAXOL  PMINOL PMINF  DPMIN
C                              TFLOWT  TMAXF TMAXOL  TMINOL TMINF  DTMIN
C                              NOTE -  FOR PH(7)  =  -1, 0  VALUES ARE
C                                      TO BE GIVEN IN DEGREES.
C                                    - FOR PH(7)  =  +1  VALUES ARE TO
C                                      BE GIVEN IN MULTIPLES OF  PIE .

C      100       142      42   ATRADS(6) CROS(6)   E(6)      TEMPD(6)
C                              TTEOOL    TEMPBS(3)  TPMODS(9)  TPNCBS(3)
C      200       144      44   RHOZRO    DOTVIN     RIEROE
C                              GOAVAZ    GREFRZ     SOPLTN
C                        CONSTS RHOZRO  BETAIR  BETAZR  RVSPLT  TEST
C      400       149     CONSTS STRTIN  VARNCE  PHCMIN  PHCMAX  SINCR
C                              AMPFIT  AMPSIN  WAVELN  FYMODE  IREFMX
C                        CONSTS AMPLOE(IREFMX)
C                        CONSTS FYMEF(IREFMX)

C     1000       103      13   ASTCIN  ATICIN  TPICIN  SNSSWT  TMMAX
C                              FASHTS  FTPOPL  FYFLWT  TPDSTS  TNPSTS
C                              PFLOWT  TFLOWT  DEGATS  ADTSTS  ACELTS
C                              AFLOWT  ACEXTS  DTSHTS  TEMPTS  TPEXTS
C                              THEXTS  TCEXTS
C     2000       103     PITCHI GREFRZ  MOS     CDZERO  CP      CPDRAG
C                              DALPHS  DALPTS  ALPRAT  PHIRAT  PHIRTE
C                              BZOL    CPMXOL  PI2SOL  PBLD    ALPHAI
C                              MODI    PATOL1  PATOL2  PDGCOL  CSCPOL
C                              PTCADJ  FYMIN   FYMAX   FYMODF  FYMIN
C                              PYMAX   FYMODF  GAINOL  PYGAIN  FGMAX
C                              SEGMMO  ALPON1  ALPON2  ALPOMI  ALPOBS
C                              EATMPY  TIMMIN  MOSUB   MONAX   MULTMD
C                              NEMPPC  MAXPPC  ALPSJN  PHISJN  GNMMAL
C                              GMAMPM  FYTMAX  FYTMAX  PEGTRE
C                              NOTE  -  EXCEPT FOR GREFRZ, THESE VALUES
C                              ARE CHANGED ONLY WITHIN SUBROUTINE PITCH
C     4000       109     PITCHI ALPHAE  MODE    ALPHAS  MODS    PHISAP
C                              ALPHAT  MODI    ICPSOL
```



```
C     COMMON STATEMENTS FOR GOPTNS VARIABLES
      COMMON /C01/     VMIN,VMIN2,VMINOL,VMINF,VMINL,DVMIN,
     1                 FMACHN,FMACH2,FMNOUT,FMLMNU,FMLMN2,FMNMOL,FMLMNF,
     2                 FMLMNL,DMACHN
      COMMON /C02/     FTPOPL,ENERGY,TENRGY,ENGOUT,ENGYOL,ENRGYF,ENRGYR,
     1                 ENRGYL,DENRGY,DEGYOL,DEGYCH,DEGCOL,ENGCON
      COMMON /C04/     FYFLWT,YMAX,YMAXF,YMAXOL,YMIN,YMINOL,YMINF,YMINL,
     1                 DYMIN
      COMMON /C010/    ACELTS,AFLOWT,ACEXTS,ACCEL,ACCEL2,AMAX,AMAX2,
     1                 AMAXF,AMAXOL,AMIN,AMIN2,AMINOL,AMINF,DAMIN,
     2                 TESTAZ,TTAZOL,ACCLBS(9),ANGLDZ,ALDLDZ,ACC2BM,
     3                 ACC2BL,ACC2RE,ACC2RF,ENOOUT,ENOR12,ENOR22,ENOR3,
     4                 ENOR42,ENOR5,ENOR6,ENOR7,ENDMAX,EDMXOL,ENDMXF,
     5                 ENDMXY
      COMMON /C020/    TIMAX,TIMAXF,TIMAXL,TIMXOL,DTIMIN,BMAX,BMAXF,
     1                 BMAXOL,DBMIN,SNEOUT,SDMAX,SDMAXF,SDMAXL,SDMXOL,
     2                 SNTERD,DSSAVE,DSMAXS,DSMAX,DSMIN,DSMIND,DSMINF,
     3                 DSMINT,DSMNGD,ADTEST,ADNEGT,AGTEST,AGTSOL,AGSTOP,
     4                 AGSPOL,AGNEGT,ADGCON
      COMMON /C040/    PFLOWT,PMAX,PMAXF,PMAXOL,PMIN,PMINOL,PMINF,DPMIN,
     1                 TFLOWT,TMAX,TMAXF,TMAXOL,TMIN,TMINOL,TMINF,DTMIN
      COMMON /C0100/   TEMPTS,TPEXTS,THEXTS,TCEXTS,TPNDEG,TANDEG,TPEORT,
     1                 TPMAX(6),ATPHTT(6),ATPLTT(6),P1(6),P2(6),PB(6),
     2                 P1CNST,P2CNST,PBCNST,ATRADS(6),CROS(6),E(6),
     3                 YOMPIN,TESTPO,TTEOOL,TEMPBS(3),TPMODS(9),
     4                 TPNCBS(3),TANNOL,TALNOL,TANODM,TANODL,TANONE,
     5                 TANCNE,TANOOF,TANCAP
      COMMON /C0200/   RHOZRO,BETAV

C     DOUBLE PRECISION STATEMENTS FOR GOPTNS VARAIBLES
      DOUBLE PRECISION SDMAX,SDMAXF,SDMAXL,SDMXOL,SNTERD
      DOUBLE PRECISION TORPIN(6)
```



```
C     DATA VALUES WHICH CAN BE ALTERED UNDER CONTROL OF GOPTNS
C     GOPTNS =      1
      DATA   VMINOL,VMINF,VMINL,DVMIN/5000.0,2000.0,2000.0,1000.0/
      DATA   FMNOUT,FMNMOL,FMLMNF,FMLMNL,DMACHN/0.2,5.0,3.0,3.0,1.0/
C     GOPTNS =      2
      DATA   FTPOPL,ENGOUT,ENGYOL,ENRGYR,ENRGYL,DEGYOL,DEGCOL,ENGCON/
     1       0.0,0.0,1000000.0,0.0,100000.0,250000.0,1.0,1.0/
C     GOPTNS =      4
      DATA   FYFLWT,YMAXF,YMAXOL,YMINOL,YMINF,YMINL,DYMIN/-1.0,
     1       1000000.0,1000000.0,1000000.0,200000.0,90000.0,50000.0/
C     GOPTNS =     10
      DATA   AFLOWT,AMAXF,AMAXOL,AMINOL,AMINF,DAMIN,TTAZOL,ACCLBS/-1.0,
     1       50.0,0.0,0.0,-5.0,10.0,1.0,-1.0,0.125,5.0,0400000000230,
     2       0400000000000,0400000000010,0400000000100,0400000000010,
     3       0400000000200/
      DATA   ENOOUT,ENOR12,ENOR22,ENOR3,ENOR42,ENOR5,ENOR6,ENOR7,EDMXOL,
     1       ENDMXF,ENDMXY/0.0,16.0,36.0,2150.0,15.0,1830.0,10.0,6.0,
     2       0.25,10.0,6.0/
C     GOPTNS =     20
      DATA   TIMAXF,TIMAXL,TIMXOL,DTIMIN/86400.0,86400.0,600.0,600.0/
      DATA   BMAXF,BMAXOL,DBMIN/62.6310531,0.523598776,0.523598776/
      DATA   SNEOUT,SDMAXF,SDMAXL,SDMXOL,SNTERD/0.0,202,008,201,000/
      DATA   DSMAXS,DSMAX,DSMIN,DSMIND,DSMINF,DSMINT,DSMNGD/324290.0,
     1       324290.0,512.0,2048.0,0,000400820125,0.5,0.000400820125/
C     DSMAXS =  DSMAX                    DSMIN  =  DSMAX / 2.00010
C     DSMIND =  DSMAX / 2.0008           DSMINF =  DSMAX / 2.00030
C     DSMINT =  DSMAX / 2.00070          DSMNGD =  DSMAX / 2.00030
      DATA   ADTEST,AGTSOL,AGSPOL,ADGCON/-1.0,25.0,1000.0,0.0400000001/
C     GOPTNS =     40
      DATA   PFLOWT,PMAXF,PMAXOL,PMINOL,PMINF,DPMIN/-1.0,6.28318531,
     1       3.14159265,-3.14159265,-6.28318531,0.785398163/
      DATA   TFLOWT,TMAXF,TMAXOL,TMINOL,TMINF,DTMIN/-1.0,12.5663706,
     1       6.28318531,-6.28318531,-12.5663706,0.785398163/
C     GOPTNS =    100
      DATA   ATRADS/2.0,1.5,1.5,1.075,1.5,1.075/,CROS/6*0.5,2.0,3.0/
      DATA   E/0.5,0.5,0.78125,0.625,0.78125,0.625/,TOMPIN/60550.000/
      DATA   P1CNST,P2CNST,PBCNST,TTEOOL/4.44831603E-09,4.7560E-13,
     1       4197.71191,0.0/
C     P1CNST      =  17609.9  /  (   53470.C3237(6)  *  26000.0093.15  )
C     P2CNST      =  4.7560E-13            PBCNST      =  P1CNST  /  P2CNST
      DATA   TEMPBS/-100.0,-1.0,-2000.0/,TPNCBS/0.0,10000.0,0.0/
      DATA   TPMODS/-300.0,-9.25,-2000.0,0400000000200,0400000000100,
     1       0400000000010,0400000000340,0400000000200,0400000000300/
C     GOPTNS =    200
C     NOTE  -  DOTVIN = BETAV00-1 IS READ IN INSTEAD
C     NOTE- FOR RIEROE, SOPLTN, GOAVAZ, GREFRZ - SEE SUBROUTINE VARIABLES
      DATA   RHOZRO,BETAV/0.002   4.23531910E-05/
C     GOPTNS =    400
C     SEE SUMMARY OF VALUES WHICH MAY BE CHANGED UNDER CONTROL OF GOPTNS
C     GOPTNS =  1000
C     NOTE  -  THESE VALUES ARE ALWAYS SET AT THE BEGINNING OF EACH RUN
C     SEE SUMMARY OF VALUES WHICH MAY BE CHANGED UNDER CONTROL OF GOPTNS
C     GOPTNS =  2000   AND   GOPTNS =  4000
C     SEE SUMMARY OF VALUES WHICH MAY BE CHANGED UNDER CONTROL OF GOPTNS
```

BLOCK DATA CCLLDA 6.0 PAGE 5

```
C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /COSSET/  DSDUMF(7),DMAX1,DMAX2,DSDUMB(1)
      COMMON /COCTAL/  OM400,O77S,O83774,O8777
      COMMON /CPTTOS/  TDENTY,PRESS,TMOL,TATM,DENSTY,SNDSPD,SDSPD2,VMOL,
     1                 YEAR,IPTOS1,IPTOS2,IPTOS3,IPTOS4
      COMMON /CRADUS/  BSHAPE,SSHAPE,REQUD,REQU2D,FYD1,FYD2,FYD3,FYD4,
     1                 FYS05,FYS06,FYS07,FYS08,FYS09,FEARTH,EEARTH,
     2                 THETA2,RZERO0,RZ2D
      COMMON /CRANGE/  GITYLE,DOWNR,CROSSR,PSIRG,PSIRGA
      COMMON /CPCTA/   PDDUMF(10),SORLTH
      COMMON /CGRAV/   GRDUMF(10),SINP2,GMD,GOZVJ0,GOZVJ2,GDAVR2,GAEPR2,
     1                 TGAPR2
      COMMON /CPITCH/  PIDUMF(20),ALPRAT,PHIRAT,PHIRTE,PIDUMB(1)

C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION REQUD,REQU2D,FYD1,FYD2,FYD3,FYD4,RZERO0,RZ2D
      DOUBLE PRECISION ONEOPD,GRAVPD,GMD,GOZVJ0,GOZVJ2,GDAVR2

C     DATA STATEMENTS FOR SUBROUTINE VARIABLES
      DATA    DMAX1,DMAX2/1.5625,O201627076000/
C     DMAX1  =  89.52469547 DEGREES  =  1.5625      RADIANS
C     DMAX2  =  89.99930137 DEGREES  =  1.57078593 RADIANS
      DATA    OM400,O77S,O83776/O400000000,O777777777,O377400000000/
      DATA    O8777/O777700000000/
      DATA    IPTOS1,IPTOS2,IPTOS3,IPTOS4/4HPECT,4HPREVUS,4HEDSPD2,
     1        4HRETURN/
      DATA    REQUD,REQU2D,FYD1,FYD2,FYD3,FYD4,FYS05,FYS06,FYS07,FYS08,
     1        FYS09,FEARTH,EEARTH,THETA2,RZERO0,RZ2D/
     2        20925839.09501312335960,437890775311722.045605,
     3        -20925839.09501312335960,0.000,0.000,0.000,70155.0235,
     4        9.04345410F+09,-117.790,-3.320,209.250599,0.00335295469,
     5        0.00001,-0.349065851,20910100.7261799420265,
     6        437967272600101.357467/
C     REQUD  =  6378196.000/3.304000  =  - FYD1      REQU2D  =  REQUD**2
C     FYS05  =               REQU       *  FEARTH
C     FYS06  =      2.0  *  REQU ** 2   *  FEARTH ** 2
C     FYS07  =  -  2.0  *  REQU        *  FEARTH ** 2  *  0.25   -   0.15
C     FYS08  =  - 3.320   THIS VALUE WAS ARRIVED AT TO ADJUST THE RESULT
C     FYS09  =               REQU       *  EEARTH
C     FEARTH =  1.0 / 298.20
C     EEARTH =  1.0 / 100000.0     THIS VALUE IS MERELY AN ESTIMATE
C     THETA2 =  - 20.0 * DMDI PIED / 180.0 )
C     RZERO0 =  DSQRT( 3.9860320+14 / 0.30480003 / 32.1700 )
C     RZ2D   =  RZERO0 * RZERO0
      DATA    DOWNR,CROSSR,PSIRG,PSIRGA/4*0.0/
      DATA    SORLTH/235.91/
      DATA    SINP2,GMD,GOZVJ0,GOZVJ2,GDAVR2,GAEPR2,TGAPR2/0.0,
     1        14076539914211F060.6697,32.1462336179000301093,
     2        -1082.4800-06,32.1700,32.17,64.34/
C     GMD    =  3.9860320+14 / 0.30480003          GOZVJ0 =  GMD / REQU2D
      DATA    ALPRAT,PHIRAT,PHIRTE/0.1,0.2,0.3/

      END
```

SUBROUTINE ICITIL 6.0 PAGE 2

```
C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /COSSET/  ANGINC,ATALIC,ATPHIC,ATPSIC,TRPINC,DLTEST,DBREQ,
     1                 DMAX1,DMAX2,DMAXC,
      COMMON /COCTAL/  OMAC0,O77S,O83774,O8777
      COMMON /CPTTOS/  TDENTY,PRESS,TMOL,TATM,DENSTY,SNDSPD,SDSPD2,VMOL,
     1                 YEAR,IPTOS1,IPTOS2,IPTOS3,IPTOS4
      COMMON /CRADUS/  BSHAPE,GSHAPE,REQUD,RADUMM(10),RZERO0,RZ2D
      COMMON /CRANGE/  GITYLE,PGDUMD(4)
      COMMON /CRDTA/   BROT,RROT,WD,W2,TW,TW2,ROTG,ROTP,ROTV,SORLTH
      COMMON /CGRAV/   BGRAV,GRAVG,GRAVP,GRAVV,ONEOPD,GRAVPD,GRAVR,
     1                 GRAVT,SINO2,GMD,GRDUMM(4),GDAVR2,GAEPR2,TGAPR2
      COMMON /CPITCH/  ATTACK,ATAKNE,ATKALP,ATKPHI,BANK,CASMAX,ROLL,
     1                 CD2FM,CDSO2M,CLSO2M,MOD,DRAG,HIPT,HIPTSP,HIPTCP,
     2                 PSIRGT,PSIRGC,CSCPOR,ALPRAT,PHIRAT,PHIRTE,BIASPR

C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION REQUD,RZERO0,RZ2D
      DOUBLE PRECISION WD
      DOUBLE PRECISION ONEOPD,GRAVPD,GMD,GDAVR2

C     EQUIVALENCE STATEMENTS FOR SUBROUTINE VARIABLES
      EQUIVALENCE     ( ,WD)

C     MISCELLANEOUS TYPE AND DIMENSION STATEMENTS FOR LOCAL VARIABLES
      REAL            CODE(2)
      DOUBLE PRECISION VH,VH2,V3,VE2,VN,VE

C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE     (AA,OUT)

C     LOCAL DATA VALUES
      DATA   O1000/O1000/

C     FORMATS
    1 FORMAT(  17H1      CASE NUMBER, I8, 5X, 14HLETTER CODE    , 2A6, 8X,
     1         9F5.0, O6, I5  /  12H0   ANGINC =, F9.6, 11H   ATTINC =,
     2         F9.6, 11H   TRPINC =, F6.2  )
    2 FORMAT(  1H 6A8, 10F9.0  )
    3 FORMAT(  12H0    PHI     =, F9.3, 10H DEGREES    CD S/M =, F11.7,
     1         20H SQ.FT./SLUG   L/D =, F11.7  )
    4 FORMAT(  1H 6A8, 10F9.0  /
     1         53H0   NO ATMOSPHERE WILL NOT BE USED FOR THIS CASE.
     2         53H ALL AERODYNAMIC FORCES WILL BE SET EQUAL TO ZERO.    )
    5 FORMAT(  23H0    INITIAL VELOCITY  =, 1PD22.10, 9H  FT/SEC., 9X,
     1         28HINITIAL GEOCENTRIC RADIUS  =, 0PD24.9,  7H  FEET      /
     2         1H0, 22X, 5HGLIDE, 37X, 10HA N G L E S  O F              /
     3         54H  VELOCITY  ALTITUDE   ANGLE   AZIMUTH LATITUDE LONGIT
     4         51HUDE   TIME   ATTACK   ROLL  NO. ACCEL  TEMP(S) CNT    )
    7 FORMAT(  1H , 104X, 11H NSTP NUS J,  4X 4H IOT )                 /
     1         23H  (FT/SEC)   (FEET)     , 4X 5H(DEG), 4X 1, 5H(SEC)
     2         35H(DEG)  (DEG)      (G'S)   (DEG-R)                     )
    8 FORMAT(  1H , 104X, 29H DOWNR   CROSSR    PSIRG                  /
     1         23H  (FT/SEC)   (FEET)     , 4X 5H(DEG), 4X 1, 5H(SEC)
     2         35H(DEG)  (DEG)      (G'S)   (DEG-R)    , 3X 4X5H(DEG) )  )
```

SUBROUTINE ICITIL 6.0 PAGE 1

```
$IBFTC ICML     FULIST,PMAP,DD,DECK
      SUBROUTINE INITIL

C     SUBROUTINE INITIL CODE NUMBER ICITIL                   VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO INITIALIZE GLOBAL AND SUBROUTINE
C     VARIABLES AT THE START OF EACH CASE

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CNSS/    NWFPTS,NSTOP
      COMMON /CONTRL/  SNSSWT,TNCNTS,GOPTNS,NOCASE,KNTMAX,TMC,SMATOT,
     1                 FRSHTS,TPOGTS,TMPSTS,DEGATS,ADTSTS,DTSHTS
      COMMON /CVAR/    V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,BETAD,
     1                 ENDUED,TEMPD,S3
      COMMON /CIVAR/   VINITD,VINITD,GDINIT,PDINIT,BDINIT,TDINIT
      COMMON /CPVAR/   FDUMF(6),FRD,FYD,FDUMM(37),RTONTV
      COMMON /CAVAR/   AVAR(32)
      COMMON /CSCAP/   ALPHAD,SADUMM(5),PHIDEG,PHIOLD,SINPHI,COSPHI
      COMMON /CINTS/   R,K,NTP,IGPDYD,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTOL,NOPTEV,IDSTOP,KOUNTP,KOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IT,ITSAVE
      COMMON /CPIES/   PIED,PIE02D,PIE720,RDDDEG,DEGRAD,RDDDEG,PDUMB(7)

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,BETAD
      DOUBLE PRECISION ENDUED,TEMPD(6),SD
      DOUBLE PRECISION VINITD,VINITD,GDINIT,PDINIT,BDINIT,TDINIT,FRD,FYD
      DOUBLE PRECISION PIED,PIE02D,PIE72D,RDDDEG,DEGRAD

C     EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
      EQUIVALENCE     (V2,V2D),(V,VD),(R,RD),(GAMMA,GAMMAD),(PSI,PSID)
      EQUIVALENCE     (DELTA,DELTAD),(THETA,THETAD),(TIME,TIMED)
      EQUIVALENCE     (BETA,BETAD),(ENDURE,ENDUED),(TEMPS,TEMPD),(S,SD)

C     COMMON STATEMENTS FOR GOPTNS VARIABLES
      COMMON /C01/     O1DUMF(7),FMACH2,O1DUMB(7)
      COMMON /C02/     FTPOPL,O2DUMB(12)
      COMMON /C04/     FYPLUT,O4DUMB(5)
      COMMON /C010/    ACELTS,APLOWT,ACERTS,ACCEL,ACCEL2,O10DUM(30)
      COMMON /C020/    O20DUF(22),DSNOT,O20DUM(5),ADTEST,ADRIGT,AGTEST,
     1                 AGTSOL,AGSTOP,ACIPOL,AGAREF,AGDCAN
      COMMON /C040/    PFLOWT,O40DUM(7),TFLOWT,O40DUB(7)
      COMMON /C0100/   TEMPTS,TPERTS,TNERTS,TCERTS,TPNDEG,TANDEG,
     1                 C100DU(95)
```

SUBROUTINE ICITIL 6.0 PAGE 3

```
C     ALLOW FOR OVERALL RUN INITILIZATION
      IF( NOCASE .GT. 0 ) GO TO 101
C     ALLOW FOR LISTING OF BINARY DECK NAMES AND THE INPUT DATA
      CALL PRGLST
C     SET UP ERROR MESSAGE COUNTER
      CALL NFPTST( NWFPTS )
C     FIND CURRENT CLOCK TIME BEFORE STARTING
      TMP    =  CLOCK( 0.0 )
C     SET UP KICK-OFF PROCEDURE
      IF( KICKED(OFF) ) GO TO 199
C     READ IN INCREMENTATION AND OUTPUT CONTROL VALUES
      GO TO 103

C     KICK-OFF PROCEDURE
  199 CALL OUTKCK
      CALL EXIT
      STOP

C     INITIALIZATION BEFORE EACH CASE
  101 READ (5,11)   NOCASE,GITYLE,SMATOT,GSHAPE,BSHAPE,BROT,RROT,BGRAV,
     1              NTP,TNCNTS,GOPTNS,TDENTY,PHIDEG,CDSOM,MOD,CODE
   11 FORMAT(  I5, 7F2.0, I2, F3.0, O6, F2.0, F8.0, 2F11.0, 2A6  )
      NTP    =  MIN0( NTP, 6 )
      ATTACK =  0.0
      ATAKNE =  0.0
      ATKALP =  0.0
      ATKPHI =  0.0
      BANK   =  0.0
      CASMAX =  0.0
      ROLL   =  0.0

  102 IF( AND( GOPTNS, O1000 ) ) 103, 100, 103
  103 READ (5,13)   AGICIN,ATICIN,TPICIN,SNSSWT,TNMAX,ENSHTS,
     1              FTPOPL,FYPLUT,TPOGTS,TMPSTS,PFLOWT,TFLOWT,DEGATS,
     2              ADTSTS,ACELTS,APLOWT,ACERTS,DTSHTS,TEMPTS,TPERTS,
     3              TNERTS,TCERTS
   13 FORMAT(  3F10.5, 10F5.0  /  10F9.0  )
      KNTMAX =  TNMAX
      IF(    BOOL(SNSSWT)    ) 104, 100, 104
  104 IF(        SNSSWT      ) 105, 107, 100
  105 PRINT 15
   15 FORMAT( 1H0 / 49H0     SENSE SWITCH TESTS ARE ACTIVE.    PLEASE KEEP
     1          53H ALL SENSE SWITCHES OFF UNLESS OTHERWISE INSTRUCTED. /
     2          24H      PLEASE PRESS START .  24H[illegible]/1H0/1H0  )
      IF( 1.0 +  SNSSWT     ) 106, 100, 100
  106 CALL PAUSE( 106 )
      GO TO 100
  107 PRINT 17
   17 FORMAT( 1H0 / 39H0     SENSE SWITCH TESTS ARE INACTIVE.   /   1H0   )
  100 IF( NOCASE ) 101, 109, 111
  109 CALL FINAL
      STOP
```

```
C     OUTPUT OF CASE INFORMATION
 111  WRITE(6, 1)  NOCASE,CONE,GITVLE,SMATOT,USHAPE,DSHAPE,GROT,BROT,
     1             BGRAV,TDENTY,THCNTS,GOPTNS,MTP,AGICIN,ATICIN,TPICIN

C     INITIALIZATION AND OUTPUT OF AERODYNAMIC COEFICIENTS
      IF(   BOOL(TDENTY)   ) 112, 114, 112
 112  WRITE(6, 2)                          SMSLUT,TENMAX,ENGNTS,
     1             PTPOPL,PYPLUT,TPOSTS,THPSTS,PPLOUT,TPLOUT,DEGATS,
     2             ABTSTS,ACELTS,APLOUT,ACEXTS,BTSHTS,TEMPTS,TPEXTS,
     3             THEXTS,TCEXTS

      IF( -CDSON ) 113, 115, 115
 113  WRITE(6, 3)  PHIDEG, CDSON, MOD
      CDSO2M =  CDSON  / 2.0
      CLSO2M =  CDSO2M * MOD
      ALPHAD =  99.99993
      CALL SNCSPM( 01131 )
      PHIRAD =  PHIDEG * DGDRAD
      SINPHI =  SIN( PHIRAD )
      COSPHI =  COS( PHIRAD )
 1131 CONTINUE
      THCNTS =  0.0
      GO TO 116
 114  MTP    =  0
      DENSTY =  0.0
      BTDNTY =  0.0
      SDSPD2 =  1.0
      FMACH2 =  10000.0
      ALPHAD =  0.0
      PHIDEG =  0.0
      HIPTCP =  0.0
      HIPTSP =  0.0
      DRAG   =  0.0
      PYPLUT =  0.0
      ACELTS =  1.0
      ACCEL2 =  0.0
      TAUNEG =  0.0
      TPUNEG =  0.0
      THCNTS =  3.0
      GO TO 121
 115  CALL PITCHI

 116  IF( TDENTY ) 117, 118, 117
 117  YEAR   =  1960.5   +   TDENTY / 2.0
      CALL DENVPI
      GO TO 121
 118  SDSPD2 =  1000000.0

C     READ IN OF DATA VALUES WHICH CAN BE ALTERED UNDER GOPTNS CONTROL
 121  IF( BOOL(GOPTNS) )         ) 122, 151, 122
 122  CALL GOPTNC
```



```
C     READ IN OF VARIABLES AND ADJUSTMENTS WHEN NECESSARY
 151  IF( AMOD( GITVLE, 2.0 )   ) 153, 152, 153
 152  READ (5,12)  VINITD,YINITD,GDINIT,PDINIT,DDINIT,TDINIT
  12  FORMAT(  2D25.0, 4D19.9  )
 153  IF(   BOOL(GSHAPE)    ) 154, 156, 154
 154  IF(         GSHAPE    ) 155, 158, 157
 155  RD     =                 YINITD
      GO TO 159
 156  RD     =  RZERO   +   YINITD
      GO TO 159
 157  CALL RADIUS
      GO TO 159
 158  RD     =  RGEOD   +   YINITD

 159  GAMMAD =  GDINIT * DGMRAD
      PSID   =  PDINIT * DGMRAD
      DELTAD =  DDINIT * DGMRAD
      THETAD =  TDINIT * DGMRAD
      ADTEST =  SIGN( ADTEST,  - DELTA )
      DLTEST =  0.0
      DHATE  =  DHATE1
      AGNFGT =  0.0
      AGTEST =  SIGN( AMAX1(AGTSEL,-2.0*THCNTS-1.0), - GAMMA )
      AGSTMP =  AGSPEL
      AGNFGT =  0.0

 161  IF( VINITD ) 162, 163, 164
 162  VD     =  DSQRT(  GDAVE2  *    4220    /  RD    )
      V2D    =  VD * VD
      GO TO 165
 163  VD     =  DSQRT(           AMD         /   RD    )
      V2D    =  VD * VD
      GO TO 165
 164  VD     =  VINITD
      V2D    =  VD * VD

 165  IF(  GROT   ) 166, 171, 166
 166  VH     =  VD  *  DCOS( GAMMAD )
      VR     =  VD  *  DSIN( GAMMAD )
      VH2    =  VH  *  VH
      VN     =  VH  *  DCOS( PSID )
      VE     =  VH  *  DSIN( PSID )  -   WD  *  RD  *  DCOS( DELTAD )
      VH2    =  VN  *  VN    +    VE  *  VE
      GAMMAD =  DSIGN(   DATAN(  VR  /  DSQRT(VH2)  ),  GDINIT  )
      VD     =  DSQRT(  VR2  +  VH2  )
      V2D    =  VD * VD
      IF( 90.0  -  ABS(PDINIT)      )  167, 168, 169
 167  PSID   =  DSIGN(  PIED,  PDINIT  )  +   DATAN(  VE  /  VN  )
      GO TO 171
 168  PSID   =  DSIGN(  PIED2D,  PDINIT  )
      GO TO 171
 169  PSID   =  DATAN(  VE  /  VN  )
```



```
C     INITILIZATION OF ROTATIONAL AND GRAVITATIONAL TERMS
 171  IF(   BOOL( GROT )    ) 172, 173, 172
 172  ROTG   =  0.0
      ROTP   =  0.0
      ROTV   =  0.0
      GO TO 176
 173  IF(   BOOL( GROT )    ) 174, 172, 174
 174  IF(         GROT      ) 175, 175, 172
 175  ROTG   =  (( SIN(GAMMA) * COS(PSI) * SIN(DELTA)
     1                        - COS(GAMMA) * COS(DELTA)   ))  *
     2             W2 * R * COS(DELTA)  /  V2
     3               -   SIN(PSI) * COS(DELTA) * TW / V
      ROTP   =  SIN(PSI)*SIN(DELTA)/COS(GAMMA) * W2*R*COS(DELTA)/V2
     1       +  (( SIN(GAMMA) / COS(GAMMA) * COS(PSI) * COS(DELTA)
     2                        + SIN(DELTA)    ))  *  TW / V
      ROTV   =  (( COS(GAMMA) * COS(PSI) * SIN(DELTA)
     1                        + SIN(GAMMA) * COS(DELTA)   ))  *
     2             W2 * R * COS(DELTA)    *  ( - 1.0 )

 176  GRAVED =  0.0
      GRAVG  =  0.0
      GRAVP  =  0.0
      GRAVV  =  0.0
      ONENRO =  1.000
      FRD    =  0.0
      FYD    =  0.0
 177  IF( BGRAV + 4.0 ) 179, 178, 181
 178  BGRAV  =  - 1.0
 179  ONENRO =     0.0
```



```
C     INITILIZATION OF OTHER MISCELLANEOUS QUANTITIES
C     OUTPUT OF HEADINGS FOR THE OUTPUT QUANTITIES
 181  CALL RANGEI
      CALL GRAVCI
 182  DO 183  IT = 1, 32
 183  ATABL(IT)    =  0.0
      DSHARQ =         AND( DSHAT, 7777    )
      ANGINC =  OR(  AND( AGICIN, 003776 ),  DSHARQ  )
      TEPINL =  OR(  AND( TPICIN, 003776 ),  DSHARQ  )
      ATPSIC =  ( ATICIN /      PHIRTE      )
      ATPSIC =  OR(  AND( ATPSIC, 003776 ),  DSHARQ  )
      ATPHIC =  ( ATICIN /      PHIRAT      )
      ATPHIC =  OR(  AND( ATPHIC, 003776 ),  DSHARQ  )
      ATALIC =  ( ATICIN /      ALPRAT      )
      ATALIC =  OR(  AND( ATALIC, 003776 ),  DSHARQ  )
      ALPRAT =  ATICIN
      PHIRAT =  DSHARQ
      INTLST =  1
      N      =  0
      ICNDOP =  0
      ILAST  =  0
      JFAIL  =  0
      KOUNT  =  - 1
      KOUNTP =  0
      KOUNTL =  1
      NUSL   =  0
      NSTP   =  0
      NSTPTT =  0
      NUSLTT =  0
      IF(   BOOL(TDENTY)   ) 105, 104, 105
 104  WRITE(6, 4)                          SMSSUT,TENMAX,ENGNTS,
     1             PTPOPL,PYPLUT,TPOSTS,THPSTS,PPLOUT,TPLOUT,DEGATS,
     2             ABTSTS,ACELTS,APLOUT,ACEXTS,BTSNTS,TEMPTS,TPEXTS,
     3             THEXTS,TCEXTS
 105  WRITE(6, 5)  VD,RD
      IF(   BOOL(GITVLE)   ) 106, 108, 106
 106  IF(         GITVLE   ) 107, 107, 108
 107  WRITE(6, 7)
      GO TO 109
 108  WRITE(6, 8)
 109  CALL BGNLST


 193  IF(   SMSSUT    ) 1901,1903,1901
 1901 IF( SWITCH(190) ) 1902,1903,1902
 1902 AA = OUT
 1903 CONTINUE


C     NORMAL RETURN FROM SUBROUTINE ICITIL
 191  RETURN
```

```
$IBFTC GCPC     FULIST,PMAP,DD,DECK
      SUBROUTINE GOPTNC
C     SUBROUTINE GOPTNC CODE NUMBER GCPTNC                    VERSION 6.0


C     THIS SUBROUTINE IS DESIGNED TO CHANGE THOSE DATA VALUES WHICH CAN
C     BE ALTERED UNDER CONTROL OF GOPTNS


C     COMMON STATEMENTS FOR GOPTNS VARIABLES
      COMMON /CO1/      VMIN,VMIN2,VMINOL,VMINF,VMINL,DVMIN,
     1                  FMACHN,FMACH2,FMNOUT,FMLMNJ,FMLMN2,FMNNOL,FMLMNF,
     2                  FMLMNL,DMACHN
      COMMON /CO2/      FTPOPL,ENERGY,TENRGY,ENGOUT,ENGYOL,ENRGYF,ENRGYR,
     1                  ENRGYL,DENRGY,DEGYOL,DEGYCH,DEGCOL,ENGCON
      COMMON /CO4/      FYFLUT,YMAX,YMAXF,YMAXOL,YMIN,YMINOL,YMINF,YMINL,
     1                  DYMIN
      COMMON /CO10/     ACELTS,AFLOWT,ACEETS,ACCEL,ACCEL2,AMAX,AMAX2,
     1                  AMAXF,AMAXOL,AMIN,AMIN2,AMINOL,AMINF,DAMIN,
     2                  TESTA2,TTA2OL,ACCLOS(9),AMPLO2,ALOLD2,ACC2DM,
     3                  ACC2OL,ACC2ME,ACC2MF,ENDOUT,ENDR12,ENDR22,ENDR3,
     4                  ENDR42,ENDR5,ENDR6,ENDR7,ENDMAX,EDMXOL,ENDRXF,
     5                  ENDRPY
      COMMON /CO20/     TIMAX,TIMAXF,TIMAXL,TIMXOL,DTIMEN,DMAX,DMAXF,
     1                  DMAXOL,DAMIN,SHROUT,SDMAX,SDMAXF,SDMAXL,SDMXOL,
     2                  SNTERO,DSSAVE,DSMAXS,DSMAX,DSMIN,DSMIND,DSMINF,
     3                  DSMINT,DSMNGO,ADTEST,ADNEGT,AGTEST,AGTSOL,AGSTOP,
     4                  AGSPOL,AGNEGT,ADGCON
      COMMON /CO40/     PFLOWT,PMAX,PMAXF,PMAXOL,PMIN,PMINOL,PMIN2,DPMIN,
     1                  TFLOWT,TMAX,TMAXF,TMAXOL,TMIN,TMINOL,TMINF,DTMIN
      COMMON /CO100/    TEMPTS,TPEETS,THENTS,TCEETS,TPMDEG,TAMDEG,TPEQRT,
     1                  TPMAX(6),ATPMTT(6),ATPLTT(6),P1(6),P2(6),PO(6),
     2                  P1CNST,P2CNST,POCNST,PTPADS(6),CMOS(6),Z(6),
     3                  TOMPIN,TESTEO,TTEOOL,TEMPBS(3),TPHOBS(9),
     4                  TPMCBS(3),TAMNOL,TALNOL,TANDOM,TANDOL,TANDMX,
     5                  TANCMX,TANDMF,TANCMF
      COMMON /CO200/    PHOZPO,BETAY

C     DOUBLE PRECISION STATEMENTS FOR GOPTNS VARIABLES
      DOUBLE PRECISION SDMAX,SDMAXF,SDMAXL,SDMXOL,SNTERO
      DOUBLE PRECISION TOMPIN(6)
```



```
C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  SMSSHT,THCNTS,GOPTNS,ODUMBC(10)
      COMMON /CVAR/    V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,BETAD,
     1                 ENDURD,TEMPD,SD
      COMMON /CPIES/   PIED,PDUMM(8),RDORGA,PDUMB(7)

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,BETAD
      DOUBLE PRECISION ENDURD,TEMPD(6),SD
      DOUBLE PRECISION PIED

C     EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
      EQUIVALENCE        (PIE,PIED)


C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CRANDS/   RADUMP(22),DZERDD,RZ2D
      COMMON /CROTA/    SROT,GROT,WD,W2,TW,TW2,ROTG,ROTP,ROTV,SDRLTN
      COMMON /CGRAV/    GRUMP(17),GDAVR2,GAPPR2,TGAPR2

C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION DZERDD,RZ2D
      DOUBLE PRECISION WD
      DOUBLE PRECISION GDAVR2

C     EQUIVALENCE STATEMENTS FOR SUBROUTINE VARIABLES
      EQUIVALENCE        (W,WD)


C     MISCELLANEOUS TYPE AND DIMENSION STATEMENTS FOR LOCAL VARIABLES
      DIMENSION          OH(3),IOS(5),PHI(7),TH(7)
      DOUBLE PRECISION BOTTIN


C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE        (AA,OUT)


C     LOCAL DATA VALUES
      DATA    BOTTIN,OH,IOS,PH,TH/0.0D0,1D0.0,100,200,0,700,0/
      DATA    O1,O2,O4,O10,O20,O40,O100,O200,O400/O1,O2,O4,O10,O20,O40,
     1        O100,O200,O400/


C     FORMATS
   24 FORMAT(  4F10.0,  4F5.0  )
   26 FORMAT(  7F10.0  )
   28 FORMAT(  7F10.0  )
   32 FORMAT(  10F5.0 / 6D10 / 10F5.0 )
   34 FORMAT(  7F10.0 / 4D10.0, F10.0 / I15, 3F5.0, F20.0 )
   36 FORMAT(  7F10.0  )
   42 FORMAT(  6F10.0 / 6F10.0 / 6F10.0 / 6D10.0 / 7F10.0 / 6D10 )
   44 FORMAT(  F20.0, 2D20.0  /  D20.0, 2F20.0  )
```



```
C     SUMMARY OF VALUES WHICH MAY BE CHANGED UNDER CONTROL OF GOPTNS

C     GOPTNS  STATEMENT FORMAT LIST OF VALUES

C        1        123        24    VMINOL   VMINF    VMINL    DVMIN
C                                  FMNNOL   FMLMNF   FMLMNL   DMACHN
C        2        126        26    FTPOPL   ENGYOL   ENRGYR   ENRGYL
C                                  DEGYOL   DEGCOL   ENGCON
C        4        128        28    FYFLUT   YMAXF    YMAXOL
C                                  YMINOL   YMINF    YMINL    DYMIN

C       10        132        32    AFLOWT   AMAXF    AMAXOL   AMINOL   AMINF
C                                  DAMIN    TTA2OL   ACCLOS(9)
C                                  ENDR12   ENDR22   ENDR3    ENDR42   ENDR5
C                                  ENDR6    ENDR7    EDMXOL   ENDRXF   ENDRPY
C       20        134        34    TIMAXF   TIMAXL   TIMXOL   DTIMEN
C                                  DMAXF    DMAXOL   DAMIN (VALUES IN DEGREES)
C                                  SDMAXF   SDMAXL   SDMXOL   SNTERO   DSMAXS
C                                  IOS(5)   ADTEST   AGTSOL   AGSPOL   ADGCON
C       30        136        36    PFLOWT   PMAXF PMAXOL  PMINOL PMINF  DPMIN
C                                  TFLOWT   TMAXF TMAXOL  TMINOL TMINF  DTMIN
C                                  NOTE -   FOR PH(7)  =  -1, 0  VALUES ARE
C                                           TO BE GIVEN IN DEGREES.
C                                       -   FOR PH(7)  =  +1  VALUES ARE TO
C                                           BE GIVEN IN MULTIPLES OF  PIE .

C      100        242        42    PTPADS(6) CMOS(6)       Z(6)       TEMPD(6)
C                                  TTEOOL    TEMPBS(3)     TPHOBS(9)  TPMCBS(3)
C      200        146        44    PHOZPO    BOT.IN        PIEPOD
C                                  GDAVR2    GAPPR2        SDRLTN
C                            CONSTS PHOZPO1  BETA1R   BETA2R   PYSPLT   TEST
C      400        149        CONSTS STATIN   VARNCE   PACMIN   CMCMAX   SINCA
C                                  AMPFIX   AMPSIN   WAVELN   PYNMBR   INCPHI
C                            CONSTS AMPLDE(IREFHI)
C                            CONSTS PYREF(IREFHI)

C     1000        103        13    AGICIN   ATICIN   TPICIN   SMSSHT   THNMAX
C                                  EMSHTS   FTPOPL   FYFLUT   TPOSTS   THPSTS
C                                  PFLOWT   TFLOWT   DEGATS   ABTSTS   ACELTS
C                                  AFLOWT   ACEETS   STSHTS   TEMPTS   TPEETS
C                                  THPETS   SCEETS
C     2000        105        PITCHI GREFR2   WOS      CDZERO   CP       CPGMAG
C                                  DALPHS   DALPTS   ALPPAT   PHIPAT   PHEATE
C                                  BEOL     CAMROL   PI2SOL   SOLD     ALPHAI
C                                  HOBI     PPTOL1   PPTOL2   PPGCOL   CSCPOL
C                                  RTEASJ   FYMIN    FYMAX    FYMNDF   FYMIN
C                                  FYMAX    FYMXDF   GAINOL   FEGMIN   FPGMAX
C                                  FEGMND   ALPP-.   ALPOM2   ALPOMI   ALPHOS
C                                  PATMPY   TIMMIN   MOSUB    NOMAX    MALTND
C                                  MINPDC   MAXPDC   ALPSUM   PHISUM   CAMMAL
C                                  GMAXPN   FYTMAX   PYTMAX   FEGTME
C                                  NOTE  -   EXCEPT FOR GREFR2, THESE VALUES
C                                  ARE CHANGED ONLY WITHIN SUBROUTINE PITCH
C     4000        105        PITCHI ALPHAE   MODE     ALPHAS   MODS     PHISMP
C                                  ALPHAI   MODI     ICASOL
```



```
C     READ IN OF DATA VALUES WHICH CAN BE ALTERED UNDER GOPTNS CONTROL
  123 IF( AND( GOPTNS, O1     ) ) 124, 1241,124
  124 READ (5,26)  VMINOL,VMINF,VMINL,DVMIN,FMNNOL,FMLMNF,FMLMNL,DMACHN
 1241 VMIN    =  VMINOL
      VMIN2   =  VMIN  *  VMIN
      FMLMNJ  =  FMNNOL
      FMLMN2  =  FMLMNJ  *  FMLMNJ

  125 IF( AND( GOPTNS, O2     ) ) 126, 1261,126
  126 READ (5,26)   FTPOPL,ENGYOL,ENRGYR,ENRGYL,DEGYOL,DEGCOL,ENGCON
 1261 ENRGYF  =  ENRGYR
      ENERGY  =  ENGYOL
      TENRGY  =  2.0 * ENERGY
      DENRGY  =  DEGYOL
      DEGYCH  =  DEGCOL

 1270 FYFLUT  =  - ABS(FYFLUT)
  127 IF( AND( GOPTNS, O4     ) ) 128, 1281,128
  128 READ (5,28)  FYFLUT,YMAXF,YMAXOL,YMINOL,YMINF,YMINL,DYMIN
 1281 YMAX    =  - FYFLUT * YMAXF        +   ( 1.0 + FYFLUT ) * YMAXOL
      YMIN    =  - FYFLUT * YMINF        +   ( 1.0 + FYFLUT ) * YMINOL


 1310 AFLOWT  =  - ABS(AFLOWT)
  131 IF( AND( GOPTNS, O10    ) ) 132, 1321,132
  132 READ (5,32)  AFLOWT,AMAXF,AMAXOL,AMINOL,AMINF,DAMIN,TTA2OL,ACCLOS,
     1 ENDR12,ENDR22,ENDR3,ENDR42,ENDR5,ENDR6,ENDR7,EDMXOL,ENDRXF,ENDRPY
 1321 AMAX    =  - AFLOWT * 10.0         +   ( 1.0 + AFLOWT ) * AMAXOL
      AMAX2   =  AMAX  *  AMAX
      AMIN    =    AFLOWT *  0.0         +   ( 1.0 + AFLOWT ) * AMINOL
      AMIN2   =  AMIN  *  AMIN
      TESTA2  =  TTA2OL
      AMPLO2  =  0.0
      ALOLD2  =  1000000.0
      ACC2DM  =  ACCLOS(4)
      ACC2OL  =  ACCLOS(5)
      ACC2ME  =  AMAX(( - ACCLOS(1), - ACCLOS(2), - ACCLOS(3), 0.0 ) ** 2
      FMNMAX  =  FMNNOL
      ENDURD  =  0.0
```

```
135  IF( AND( GOPTNS, 020     ) ) 134, 1341,134
134  READ (5,34)  TIMAMP,TIMASL,TIMBOL,DTIMIN,DM,SDMAMP,SDMASL,SDMBOL,
     1             SNTERO,DSMAXS,IDS,ADTEST,ADTSOL,ADSPOL,ADSCON
     DMAMP   =   DM(1) * 0
     DMASL   =   DM(2) * 0
     DMBOL   =   DM(3) * 0
     DSMIN   =   DSMAXS / 2.0**IDS(1)
     DSMIND  =   DSMAXS / 2.0**IDS(2)
     DSMAMP  =   DSMAXS / 2.0**IDS(3)
     DSMINT  =   DSMAXS / 2.0**IDS(4)
     DSMBOL  =   DSMAXS / 2.0**IDS(5)
1341 TIMAX   =   TIMBOL
     TIMEO   =   0.0
     DMAX    =   DMBOL
     DFTAD   =   0.0
     SDMAX   =   SDMBOL
     DSMAX   =   DSMAXS
     DSSAVE  =   DSMAXS
     SO      =   0.0

1390 PFLOUT  =   - ABS(PFLOUT)
     TFLOUT  =   - ABS(TFLOUT)
135  IF( AND( GOPTNS, 040     ) ) 136, 140, 136
136  READ (5,36)  PH, TH
     PFLOUT  =   PH(1)
     TFLOUT  =   TH(1)
     IF( PH(7)  ) 137, 138, 139
137  PHAMP   =   PH(2) * 0
     THAMP   =   TH(2) * 0
     PHASOL  =   PH(3) * 0
     THASOL  =   TH(3) * 0
138  PHINOL  =   PH(4) * 0
     THINOL  =   TH(4) * 0
     PHINP   =   PH(5) * 0
     THINP   =   TH(5) * 0
     DPHEN   =   PH(6) * 0
     DTHEN   =   TH(6) * 0
     TH(7)   =   TH(7) * 0
     GO TO 140
139  PHAMP   =   PH(2) * PIE
     PHASOL  =   PH(3) * PIE
     PHINOL  =   PH(4) * PIE
     PHINP   =   PH(5) * PIE
     DPHEN   =   PH(6) * PIE
     THAMP   =   TH(2) * PIE
     THASOL  =   TH(3) * PIE
     THINOL  =   TH(4) * PIE
     THINP   =   TH(5) * PIE
     DTHEN   =   TH(6) * PIE
     TH(7)   =   TH(7) * 0
140  PHAX    =   - PFLOUT *       PIE   *   ( 1.0 + PFLOUT ) * PHASOL
     THAX    =   - TFLOUT * 2.0 * PIE   *   ( 1.0 + TFLOUT ) * THASOL
     PHIN    =     PFLOUT *       PIE   *   ( 1.0 + PFLOUT ) * PHINOL
     THIN    =     TFLOUT * 2.0 * PIE   *   ( 1.0 + TFLOUT ) * THINOL
```



```
141  IF( AND( GOPTNS, 0100    ) ) 142, 1421,142
142  READ (5,42)  RTRADS,CNDS,E,TEMPIN,TTEQBL,TEMPOS,TPNBOS,TPNCOS
1421 TESTEO      =  TTEQBL
     TPEONT      =  0.0
     TAMPOL      =  0.0
     TALVOL      =  1.0020
     TAMBOS  =  TPNBOS(4)
     TAMBSL  =  TPNBOS(5)
     TAMBNZ  =  AMAX1( - TPNBOS(1), - TPNBOS(2), - TPNBOS(3), 0.0 ) ** 4
     TAMCNZ  =  AMAX1( - TPNCOS(1), - TPNCOS(2), - TPNCOS(3), 0.0 ) ** 4
     TAMCPF      =  1.0020
     DO 143  IT = 1, 4
     ATPHTT(IT)  =   1.0
     ATPLTT(IT)  =   0.0
     P1(IT)      =   P1CNST  /  (   RTRADS(IT) * CNDS(IT)   )
     P2(IT)      =   P2CNST  /  (                CNDS(IT)   )  *  E(IT)
     P0(IT)      =   0.0
     IF( E(IT)   ) 1422,1423,1422
1422 P0(IT)      =              (   RTRADS(IT) * E(IT)      ) ** (-0.25)
1423 TPMAX(IT)   =   AMAX1( - TEMPOS(1), - TEMPOS(2), - TEMPOS(3), 0.0 )
143  TEMP3(IT)   =   TEMPIN(IT)

145  IF( AND( GOPTNS, 0200    ) ) 146, 1461,146
146  READ (5,46)  DWOZRO,DOTVIN,OZEROD,GDAVRZ,DREFAZ,SDELTA
     DETAV   =   4.2553191E-05
1461 DZRO    =   OZEROD * DZEROD
     TDEPRZ  =   2.0 * DREFAZ
     WO      =   2.0D0  *  PIEO  /  ( 86400.0D0  -  SDELTA )
     TW      =   2.0  *  W
     W2      =   W*W
     TW2     =   2.0  *  W2
     IF(       DOTVIN       ) 148, 148, 147
147  DETAV   =   DABS( 1.0D0 / DOTVIN )

148  IF( AND( GOPTNS, 0400    ) ) 149, 151, 149
149  CALL CONSTS

C      RETURN TO INITIALIZATION PROGRAM SUBROUTINE INITIL
151  RETURN

     END
```

9


```
$IBFTC RCGA    FULIST,MAP,DD,DECK
      SUBROUTINE RUNGEU

C     SUBROUTINE RUNGE  CODE NUMBER RCNGT                  VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO CALCULATE THE INCREMENTS TO BE ADDED
C     TO THE OLD VALUES OF THE DEPENDENT VARIABLES AS WELL AS
C     CONTROLLING THE STEP SIZE

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  SUSSUT,DUMMY(12)
      COMMON /COTCTL/  IOTCTL(151)/COTPIL/IOTPIL(50)/COTP2L/IOTP2L(50)
      COMMON /CVAR/    V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,DETAD,
     1                 ENDURO,TEMPO,SD
      COMMON /CFVAR/   FV2D,FVD,FRD,FVD,FTPO,FGD,FPD,FDD,FTD,FTID,FBD,
     1                 FED,FV2P15,RTOHTV
      COMMON /CAVAR/   AV2D,ARD,AGD,APD,ADD,ATD,ATID,ABD,ABD,ATPD,DSD
      COMMON /CDVAR/   DV2(4),DR(4),DG(4),DP(4),DD(4),DT(4),DTI(4),
     1                 DB(4),DE(4),DTP(6,4)
      COMMON /CSCGPD/  FGOLD,SING,COSG,FPOLD,SINP,COSP,FDOLD,SIND,COSD
      COMMON /CSCAP/   SAQUPE(A),SINPHI,COSPHI
      COMMON /CINTS/   N,M,NTP,ICPOYD,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NPRTEV,IOSTOP,KOUNTF,KOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IT,IITSAVE
      COMMON /CPIES/   PDUM(10),RDDOG,PDUM(7)

C     TYPE STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,DETAD
      DOUBLE PRECISION ENDURO,TEMPO(A),SD,FED
      DOUBLE PRECISION FV2D,FVD,FRD,FVD,FTPO,FGD,FPD,FDD,FTD,FTID,FBC
      DOUBLE PRECISION AV2D,ARD,AGD,APD,ADD,ATD,ATID,ABD,ABD,ATPD(6),DSD
      REAL             TEMPS(2,6),ATPS(2,6)

C     EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
      EQUIVALENCE      (V2,V2D),(V,VD),(R,RD),(GAMMA,GAMMAD),(PSI,PSID)
      EQUIVALENCE      (DELTA,DELTAD),(THETA,THETAD),(TIME,TIMED)
      EQUIVALENCE      (BETA,DETAD),(ENDUR,ENDURO),(TEMPS,TEMPO),(S,SD)
      EQUIVALENCE      (FV2,FV2D),(FV,FVD),(FP,FPD),(FV,FVD),(FTP,FTPD)
      EQUIVALENCE      (FG,FGD),(FP,FPD),(FD,FDD),(FT,FTD),(FTI,FTID)
      EQUIVALENCE      (AV2,AV2D),(AR,ARD),(AG,AGD),(AP,APD),(AD,ADD)
      EQUIVALENCE      (AT,ATD),(ATI,ATID),(AC,ABD),(AE,AED),(ATPS,ATPD)
      EQUIVALENCE      (DS,DSD),(FR,FBD),(FE,FED)
```



```
C     CALCULATION PRELIMINARIES BEFORE EVALUATION OF THE DERIVATIVES
  201 DO  337  M  =  1, 5
      IF( M -  1 +  ICPOYD ) 202, 337, 203
  202 ICPOYD =  ICPOYD  -  100
      IF( 200   +  ICPOYD ) 271, 211, 337
  203 FGA    =  GAMMAD   +  B(M)*DG(M-1)
      FPD    =  PSID     +  B(M)*DP(M-1)
      FDD    =  DELTAD   +  B(M)*DD(M-1)
      FTD    =  THETAD   +  B(M)*DT(M-1)
      FV2D   =  V2D      +  B(M)*DV2(M-1)
      FTID   =  TIMED    +  B(M)*DTI(M-1)
      FBTDAD =              B(M)*DB(M-1)
      FV     =  SQRT(FV2)

  211 CALL SNCSGA( 02111 )
      SING   =  SIN( FG )
      COSG   =  COS( FG )
 2111 CONTINUE
  212 IF(  COSG  ) 213, 213, 221
  213 WRITE(6,214) FG,FG
  214 FORMAT(1H0/47H    FLIGHT PATH ANGLE IS TOO LARGE - FG =F16.6,D16)
      ICNTRL =  214
      GO TO 710

  221 CALL SNCSPS( 02211 )
      SINP   =  SIN( FP )
      COSP   =  COS( FP )
 2211 CONTINUE

  231 CALL SNCSDE( 02311 )
      SIND   =  SIN( FD )
      COSD   =  COS( FD )
 2311 CONTINUE
  232 IF(  COSD  ) 233, 233, 239
  233 WRITE(6,234) FD,FD
  234 FORMAT(1H0/30H    LATITUDE IS OVER THE POLES - FD =    F16.6,D16)
      ICNTRL =  234
      GO TO 710
  239 SIND2  =  SIND * SIND

  241 IF(  BANK  ) 242, 251, 242
  242 CALL RANGE
      PSIRGA =  ABS( PSIRG + RDDOGA )
      BIASPA =  0.5
```



```
C     COMMON STATEMENTS FOR GOPINS VARIABLES
      COMMON /CO1N/    D1DUM(4),ACCEL2,D1DUM(27),ENDR12,DUMR22,ENDR3,
     1                 [illegible]
      COMMON /CO2N/    D2DUM(10),SDMAX,D2DUM(8),DSSAVE,DSMAXS,DSMAX,
     2                 DSMIN,DSMIND,DSMINF,D2DUM(10)
      COMMON /CO1PO/   P1DUM(25),P1(6),P2(6),D1DUM(4)
      COMMON /CO2PO/   DUM(20),DFTAV

C     DOUBLE PRECISION STATEMENTS FOR GOPINS VARIABLES
      DOUBLE PRECISION SDMAX

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /COSSET/  ANGINC,ATALIC,ATPHIC,ATPSIC,TEPINC,DSDUM(9)
      COMMON /CRCTAC/  [illegible]
      COMMON /CPTTNS/  TSEATY,PTDUM(3),DENSTY,PTDUM(8)
      COMMON /CPADUS/  DSHAPE,GSHAPE,REQUD,REQU2D,FVS1,FVS2,FVS3,FVS4,
     1                 [illegible]
     2                 THETA1,D2EQ2N,D220
      COMMON /CRANGE/  CITYLT,CRANG,CONSID,PSIRG,PSIRGA
      COMMON /CROTA/   [illegible]
      COMMON /CGRAV/   [illegible]
     1                 [illegible]
     2                 TCAFO2
      COMMON /CPITCH/  ATTACK,ATBANK,ATLALP,ATEPHI,BANK,CDSMAL,BRLL,
     1                 [illegible]
     2                 [illegible]

C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION [illegible]
      DOUBLE PRECISION WD
      DOUBLE PRECISION [illegible]

C     EQUIVALENCE STATEMENTS FOR SUBROUTINE VARIABLES
      EQUIVALENCE      [illegible]
      EQUIVALENCE      [illegible]
      EQUIVALENCE      [illegible]
      EQUIVALENCE      (W,WD)
      EQUIVALENCE      [illegible]
      EQUIVALENCE      [illegible]

C     MISCELLANEOUS TYPE AND DIMENSION STATEMENTS FOR LOCAL VARIABLES
      DOUBLE PRECISION [illegible]
      DIMENSION        B(4)

C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE      [illegible]
      EQUIVALENCE      [illegible]
C     LOCAL DATA VALUES
      DATA   [illegible]
      DATA   [illegible]
      DATA   [illegible]
```



```
  251 IF(   ABS(DSHAPE)    ) 254, 252, 254
  252 IF( M     -       1  ) 253, 261, 253
  253 FV     =  FVP     +   [illegible]
      FR     =  FV      +   [illegible]
      GO TO 261
  254 IF( M     -       1  ) 255, 256, 255
  255 FRR    =  RD      +   [illegible]
  256 IF(         SINOPE    ) 257, 258, 25[illegible]
  257 FVS04  =  [illegible] - THETA1)**2 - 0.5 ))*COSD
      FVD    =  FPD     +   FVD1   +   FVD4
      GO TO 261
  258 FVS02  =  FVS05   *  SIND2
      FVS0C  =  (( FVS06 / FR  +  FVS07  +  FVS08 * ( SIND2 - 0.5 )   ))
     1                        * ( SIND * COSD ) ** 2
      FVS04  =  [illegible] - THETA1)**2 - 0.5 ))*COSD
      FCD    =  FRD     +   FVD1   +   FVD2   +   FVD3   +   FVD4
      GO TO 261
  259 IF( 2.C     - DSHAPE ) 2501,2503,2502
 2501 FCD    =  FRD     -   [illegible]
      GO TO 261
 2502 FVD    =  FRD     -   [illegible]
      GO TO 261
 2503 FVSAP  =  FVS05   *  SIND2
      FVS03  =  (( FVS06 / FR  +  FVS07  +  FVS08 * ( SIND2 - 0.5 )   ))
     1                        * ( SIND * COSD ) ** 2
      FVD    =  FRD     +   FVD1   +   FVD2   +   FVD3

  261 IF(   ABS(TDENTY)   ) 262, 263, 262
  262 IF(         TSEATY   ) 263, 264, 263
  263 CALL [illegible]
      IF( ATTACK +  COSMAE ) 265, 266, 265
  264 DENSTY =  [illegible] * EXP( - FV * BETAV )
      IF( ATTACK +  COSMAE ) 265, 266, 265
  265 CALL PITCH
  266 WIFT   =  CLSO2H  *  DENSTY
      WIFTSP =  WIFT    *  SINPHI
      WIFTCP =  WIFT    *  COSPHI
      DRAG   =  CDSO2H  *  DENSTY  *  FV2
      ACCEL2 =  ( WIFT * FV2 / [illegible] ) ** 2  +  ( DRAG / [illegible] ) ** 2
      IF(  [illegible]  ) 267, 266, 267
  267 IF(  ICPOYD + NTP  ) 268, 269, 268
  268 FV2P15 =  FV2  *  FV2**0.075
      RTOHTV =  SQRT( DENSTY )
  269 IF( ICPOYD ) 411, 271, 411
```

10


```
271  IF( BOOL( BROT )   ) 272, 281, 272
272  IF(       BROT     ) 273, 274, 275
273  ROTG  =                               -  SINP*COSD*TWOFV
     ROTP  =                     (SIND+SING/COSG*CPSP*COSD)*TWOFV
     GO TO 281
274  W2FRCO =  W2 * FR * COSD
     WRDOV2 =  W2FRCO  /  FV2
     CPSD   =  COSP * SIND
     TWOFV  =  TW  /  FV
     ROTG   =  (SING*CPSD - COSG*COSD)*WRDOV2  -  SINP*COSD*TWOFV
     ROTP   =  SINP*SIND/COSG*WRDOV2 + (SIND+SING/COSG*CPSP*COSD)*TWOFV
     ROTV   =  ( - SING*COSD  -  COSG*CPSD )  *  W2FRCO
     GO TO 281
275  W2FRCO =  W2 * FR * COSD
     WRDOV2 =  W2FRCO  /  FV2
     CPSD   =  COSP * SIND
     ROTG   =  (SING*CPSD - COSG*COSD)*WRDOV2
     ROTP   =  SINP*SIND/COSG*WRDOV2
     ROTV   =  ( - SING*COSD  -  COSG*CPSD )  *  W2FRCO

281  IF(  BOOL(OGRAV )  ) 283, 282, 283
282  GRAVRD =  GOAVN2  *  ( R22D / ( FRD * FRD ) )
     GO TO 296
283  IF(          OGRAV     ) 284, 287, 287
284  IF( - OGRAV  -   2.0 ) 285, 286, 301
285  GRAVRD =  GMD  /  ( FRD * FRD )
     GO TO 296
286  RD2OR2 =  REQU2D  /  ( FRG * FRD )
     GOAVJO =  GO2VJO  *  RD2OR2
     GRAVJ2 =  GR2VJ2  *  RE2OR2
     GRJ2SN =  GRAVJ2 * ( 4.5*SIND2  -  1.5 )
     GRAVRD =  GOAVJO  *  ( 1.0D0  +  GOJ2SN )
     GRAVD  =  GRAVJO  *  3.0  *  GRAVJ2  *  SIND * COSD
     GRAVG  =  - GRAVD - SING * COSP / FV2
     GRAVP  =  - GRAVD * SINP / COSG / FV2
     GRAVV  =    GRAVD * COSG * COSP
     GO TO 296

287  CALL GRAVCS
     IF(        -OGRAV     ) 288, 291, 284
288  GRAVG  =  - GRAVD * SING * COSP / FV2
     GRAVP  =  - GRAVD * SINP / COSG / FV2
     GRAVV  =    GRAVD * COSG * COSP
     GO TO 296

291  GRAVG  =  - ( GRAVD * SING * COSP  +  GRAVT * SING * SINP )  / FV2
     GRAVP  =    ( GRAVT * COSP  -  GRAVD * SINP )  /  COSG  /  FV2
     GRAVV  =    ( GRAVD * COSG * COSP  +  GRAVT * COSG * SINP )

296  GOAVRG =  GRAVRD / FV2D  -  ONEORD / FRD
```



```
C     CALCULATION OF THE  .IVATIVES
301  OG(N)  =  ( -  ' .CP  -  GRAVRG  *  COSG
    1                 +  .RAVG    +   ROTG                        )  * DS
     CGCFR  =  COSG  /  FR
     GPORCD =  CGOFR  /  COSD  *  SINP
     OP(N)  =  (     NIFTSP  /  COSG
    1                 +  GPORCD  *  SIND   +  GRAVP  +  ROTP      )  * DS
     OD(N)  =           CGCFR  *  COSP                               * DS
     IF(  NTP      3C2, 304, 302
302  DO  303  *    1, NTP
30,  DTP(IT,N)    =   (      FV2P15 * RTONTY * P1(IT)    -
    1 (( (TEMPS(1,IT)+B(N)*DTP(IT-9,N))**2))**2 / FV * P2(IT)  )  * DS
304  IF( N + IABS(IGNORE)  - 1 ) 331, 305, 331

C     DS SETTING
305  ANGMAX =  AMAX1( ABS( OG(1) ),  ABS( OP(1) ),  ABS( OD(1) )    )
     ANGMAX =  OR(  AND( ANGMAX, O83774 ),  OM400     )
     DSMINA =  ANGINC / ANGMAX
306  IF( ATARNE ) 307, 321, 307
307  ATTMAX =          AND( FV,      O83774 )
311  IF( AND( ATARNE, OM2000 ) ) 316, 312, 316
312  IF( AND( ATARNE, OM40   ) ) 317, 313, 317
313  IF( AND( ATARNE, OM100  ) ) 318, 314, 318
314  WRITE(6,315) ATTACK,ATARNE,IGNORE,IGPOYO,ATTACK,ATARNE
315  FORMAT( 1H0/33H0      INVALID VALUE OF ATARNE  =,2F10.5,2I10,2O19)
     ICNTRL =  315
     GO TO 719
316  DSMINA =  AMIN1( ATPSIC * ATTMAX, DSMINA )
     GO TO 321
317  DSMINA =  AMIN1( ATPHIC * ATTMAX, DSMINA )
     GO TO 321
318  DSMINA =  AMIN1( ATALIC * ATTMAX, DSMINA )
321  IF(  NTP   ) 322, 323, 322
322  TEPMAX =  AMAX1( DTP, IT, 1, NTP, 1 )
     TEPMAX =  OR(  AND( TEP   , O83774 ),  OM400    )
     DSMINA =  AMIN1( TEPINC / TEPMAX, DSMINA )
323  DS     =  AMIN1(  AMAX1( DSMINA, DSMIN ),  DSSAVE,  DSMAX )
324  IF( SO + DSO - SOMAX ) 325, 325, 328
325  OG(1)  =  OG(1) * DS
     OP(1)  =  OP(1) * DS
     OD(1)  =  OD(1) * DS
     IF(  NTP    ) 326, 331, 326
326  DO  327  IT = 1, NTP
327  DTP(IT,1)    =  DTP(IT,1) * DS
     GO TO 331
328  IF( DSMINF - DS ) 329, 329, 325
329  NOPTEV =  320
     DS     =  0.5 * DS
     DSSAVE =  AMAX1( 0.5 * DS,  DSMINF )
     IDSTOP =  1
     NUSL   =  NUSL  +  1
     IPTF1L(NUSL)     =   NOPTEV
     IPTF2L(NUSL)     =   COUNT
     GO TO 324
```



```
331  DV2(N) =  (       GRAVP * SING    -    DRAG
    1                +  GRAVV   +   ROTV   )  *      2.0              * DS
     DT(N)  =           GPORCD                                        * DS
     DR(N)  =        -  SING                                          * DS
     DTI(N) =           1.0/FV                                        * DS
     DB(N)  =           CGOFR                                         * DS
332  IF( ENDR12 -  ACCEL2 ) 333, 336, 336
333  IF( ACCEL2 -  ENDR22 ) 334, 334, 335
334  DE(N)  =  ( ACCEL2 - ENDR42 )  /  ENDR3  /  FV                   * DS
     GO TO 337
335  DE(N)  =  ENDR6 ** ( SQRT(ACCEL2)/ENDR7 )   /  ENDR5  /  FV      * DS
     GO TO 337
336  DE(N)  =  0.0
337  IGPOYO =  0

C     INTEGRATION BY A FOURTH ORDER RUNGE-KUTTA METHOD
351  AG     =  ( DG(1)  + (DG(2) +DG(3) )*2.0 + DG(4)    ) * ONEOSR
     AP     =  ( DP(1)  + (DP(2) +DP(3) )*2.0 + DP(4)    ) * ONEOSR
     AO     =  ( DO(1)  + (DO(2) +DO(3) )*2.0 + DO(4)    ) * ONEOSR
     AV2    =  ( DV2(1) + (DV2(2)+DV2(3))*2.0 + DV2(4)   ) * ONEOSR
     AR     =  ( DR(1)  + (DR(2) +DR(3) )*2.0 + DR(4)    ) * ONEOSR
     AT     =  ( DT(1)  + (DT(2) +DT(3) )*2.0 + DT(4)    ) * ONEOSR
     ATI    =  ( DTI(1) + (DTI(2)+DTI(3))*2.0 + DTI(4)   ) * ONEOSR
     AB     =  ( DB(1)  + (DB(2) +DB(3) )*2.0 + DB(4)    ) * ONEOSR
     AE     =  ( DE(1)  + (DE(2) +DE(3) )*2.0 + DE(4)    ) * ONEOSR
     IF(  NTP    ) 352, 360, 352
352  DO  353  IT = 1, NTP
353  ATPS(1,IT)=(DTP(IT,1)+(DTP(IT,2)+DTP(IT,3))*2.+DTP(IT,4)) * ONEOSR


360  IF(    SNSSWT     ) 3601,3603,3601
3601 IF( SWITCH(360) ) 3602,3603,3602
3602 AL = OUT
3603 CONTINUE

C     NORMAL RETURN FROM SUBROUTINE RUNGKU
361  RETURN

C     SPECIAL RETURN AFTER CALLING RUNGKU AT STATEMENT NUMBER 407
411  RETURN

C     SPECIAL ERROR RETURN FROM SUBROUTINE RUNGKU
719  JFAIL  =  JFAIL   +   1
     IOTCTL(JFAIL)  =  ICNTRL
     ILAST  =  1
     DO  720  IT = 1, 16
720  ATPS(1,IT-9)   =  0.0
     RETURN

     END
```

```
$IBFTC CCAA    FULIST,PMAP,SO,DECK
       SUBROUTINE CHECAA

C      SUBROUTINE CHECAA CODE NUMBER CCECAA                    VERSION 6.0

C      THIS PROGRAM IS DESIGNED TO CHECK THE PROGRESS OF THE AZIMUTH
C      CHANGE AND TO PREVENT IT FROM INCREASING BEYOND 90 DEGREES BY
C      REDUCING THE BANK ANGLE TO ZERO DEGREES

C      COMMON STATEMENTS FOR GLOBAL VARIABLES
       COMMON /CPVAR/    FOUR(2),FV,FOUR(23)
       COMMON /CAVAR/    AOUR(30),OS,AOUR(1)
       COMMON /CSCAR/    SADUMP(7),COSPHI
       COMMON /CINTS/    M,N,NTP,IGPOTO,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
      1                  ICNTRL,NMPTEV,IOSTOP,KOUNTP,KOUNTL,NSTP,NUSL,
      2                  NSTPTT,NUSLTT,IT,ITSAVE
       COMMON /CPIPS/    POUR(2I),PIE02,POUR(7),RODEGR,DEGRDR,PIE02L,
      1                  POUR(15)

C      COMMON STATEMENTS FOR GOPINS VARIABLES
       COMMON /CO20/     O20DUP(25),OSQINP,O20DUB(10)

C      COMMON STATEMENTS FOR SUBROUTINE VARIABLES
       COMMON /COSSET/   ANGINC,OSCUMM(3),DSREQ,DSCUMM(3)
       COMMON /CPTTOS/   PTNUMP(4I),DENSTY,PTDUM8(8)
       COMMON /CRANGE/   GITYLE,DCUMM,CROSSR,PSIRG,PSIRGA
       COMMON /CPITCH/   ATTACK,ATAEME,ATKALP,ATKPHI,BANK,CRSMAX,ROLL,
      1                  CNSDM,CD3D2M,CL3D2M,MOD,DRAG,MIPT,MIPTSP,MIPTCP,
      2                  PSIRGT,PSIRAC,CSCPDM,ALPRAT,PHIRAT,PHIRTE,BIASPR
```



```
  421  IFI  ROLL  ) 422, 449, 422
  422  IFI-PSIRGT ) 423, 424, 428
  423  IFI PSIRGT -  PSIRGA -  BIASPR ) 424, 440, 440
  424  IFI ORI ATTACK, CRSMAX )  ) 425, 427, 425
  425  CSCPDM =  ( COSPHI - 1.0 ) / PHIRTE * CL3D2M
       IFI ATTACK +  CRSMAX ) 427, 426, 427
  426  ATTACK =  0.0
       CRSMAX =  0.0
  427  PSIRGC =  PIE02 + CSCPDM * DENSTY * FV
  428  IFI PSIRGC -  PSIRGA -  BIASPR ) 429, 440, 440
  429  IFI                   BIASPR ) 431, 432, 431
  431  CALL RANGE
       PSIRGA =  ABSI PSIRG * RODEGR )
  432  BIASPR =  ANGINC
       IFI-PSIRGT ) 433, 435, 435
  433  IFI PSIRGT -  PSIRGA ) 434, 435, 435
  434  PSIRGT =  0.0
  435  IFI PSIRGC -  PSIRGA ) 436, 437, 440
  436  NMPTEV =  436
       IFI OSMINP - OS ) 601, 437, 437
  437  IFI PSIRGA -  PIE02L ) 796, 797, 797

C      NORMAL RETURN FROM SUBROUTINE CHECAA
  440  RETURN

C      POINT CHECK CONTROL IMPLEMENTATION

C      NO POINT EVALUATION - OUTPUT CHECK CONTROL NOT PASSED
  601  IGNORE =  1
  602  DSOFF  =  0.5
       OS     =  DS +  OSREQ
  603  IGPOTO =  0

C      SPECIAL RETURN FOR NO POINT EVALUATION
  604  RETURN

C      OUTPUT CONTROL VALUES PRE-EVALUATION
  796  ICNTRL =  796
       ATKPHI =  11.0
       BANK   =  ABSI SIGNI 1.0, ROLL ) +  1.0 )
       CSCPDM =  0.0
       PSIRGC =  PIE02
       PSIRGT =  - PIE02
       GO TO 801
  797  ICNTRL =  797
       ATKPHI =  SIGNI 12.0, ATKPHI )
       BANK   =  0.0
       ROLL   =  0.0

C      SPECIAL RETURN FOR OUTPUT OF INTERMEDIATE OR FINAL VALUES
  801  IGPOTO =  - IGPOTO
       RETURN

       END
```

12


```
*DECK CCKO    PULIST,PMAP,DD,DECK
      SUBROUTINE CHECKO

C     SUBROUTINE CHECKO CODE NUMBER CCECKO                    VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO CHECK THE VALUES FOR THE VEHICLES
C     ENERGY, MACH NUMBER, VELOCITY, HEIGHT, AND ENDURANCE FACTOR FOR
C     THE PILOT  -  IF ANY CHECK SHOWS THAT AN INTERMEDIATE OR FINAL
C     OUTPUT IS DESIRED, THE PROPER CONTROL VARIABLES WILL BE SET

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  ODUMP(7),FRSHTS,ODUMB(5)
      COMMON /CTVAR/   VZTEMP,TDUMM(15),ETEMP,TDUMB(3)
      COMMON /CFVAR/   FDUMP(6),FY,FDUMB(5)
      COMMON /CAVAR/   ADUMP(30),DS,ADUMB(1)
      COMMON /CINTS/   M,N,NTP,IGPOYD,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NOPTEV,IDSTOP,KOUNTF,KOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IT,ITSAVE

C     COMMON STATEMENTS FOR GOPTNS VARIABLES
      COMMON /CO1/     VMIN,VMIN2,VMINOL,VMIGP,VMINL,DVMIN,
     1                 FMACHM,FMACH2,FMNOUT,FMLMNU,FMLMN2,FMNNOL,FMLMNF,
     2                 FMLMNL,DMACHM
      COMMON /CO2/     FTPOPL,ENERGY,TENRGY,ENGCUT,ENGTOL,ENRGTF,ENRGVR,
     1                 ENRGTL,DENRGY,DEGYOL,DEGYCH,DEGCOL,ENGCON
      COMMON /CO4/     FYFLUT,YMAX,YMAXF,YMAXOL,YMIN,YMINOL,YMINF,YMINL,
     1                 DYMIN
      COMMON /CO19/    O1DDUF(31),FNDOUT,ENDR12,ENDR22,ENDR3,ENDR42,
     1                 ENDR5,ENDR4,ENDR7,ENDMAX,ENDRCL,ENDREF,ENDRPY
      COMMON /CO20/    O2DDUF(25),DSMINF,O2DDUB(10)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CMSPT/   DSDUMP(6),DSFED,DSDUMB(3)
      COMMON /CGRAV/   GRDUMP(20),TGAPRI
```



```
C     CHECKS ON THE ENERGY, MACH NUMBER, VELOCITY, AND HEIGHT
 500  IF( FRSHTS ) 551, 502, 501
 501  IF( FTPOPL ) 5011,5012,510
 5011 IF( - TGRPRI*FY - VZTEMP  +  TENRGY ) 502, 701, 551
 5012 IF( - VZTEMP/TGAPRI - FY  +  ENERGY ) 502, 701, 551
 502  IF( FMLMN2 -  FMACH2 ) 503, 702, 552
 503  IF( VMIN2  -  VZTEMP ) 504, 703, 553
 504  IF(   FY   -    YMAX  ) 505, 704, 554
 505  IF(  YMIN  -     FY   ) 5050,705, 555
 5050 IF(     - FYFLUT      ) 5051,506, 5055
 5051 IF( -  YMAX  +  FY  +   DYMIN  ) 5052,5053,5053
 5052 YMAX   =  YMAX   -   DYMIN
      GO TO 5051
 5053 IF( -  FY  +  YMIN  +   FYMIN  ) 5054,506, 506
 5054 YMIN   =  YMIN   +   DYMIN
      GO TO 5055
 5055 IF( KOUNTL -  KOUNT  ) 5056,506, 506
 5056 IF( IGNORE ) 506, 5057,506
 5057 FYFLUT =  1.0
      GO TO 5051
 506  IF( ETEMP  -  ENDMAX ) 510, 706, 556

C     NORMAL RETURN FROM SUBROUTINE CHECKO
 510  RETURN

C     TRANSFER CONTROL FOR CHECKS ON THE TEMPORARY VALUES
 551  NOPTEV =  551
      IF( DSMINF - DS ) 601, 701, 701
 552  NOPTEV =  552
      IF( DSMINF - DS ) 601, 702, 702
 553  NOPTEV =  553
      IF( DSMINF - DS ) 601, 703, 703
 554  NOPTEV =  554
      IF( DSMINF - DS ) 601, 704, 704
 555  NOPTEV =  555
      IF( DSMINF - DS ) 601, 705, 705
 556  NOPTEV =  556
      IF( DSMINF - DS ) 601, 706, 706

C     POINT CHECK CONTROL IMPLEMENTATION

C     NO POINT EVALUATION - OUTPUT CHECK CONTROL NOT PASSED
 601  IGNORE =  1
 602  DSRED  =  0.5
      DS     =  DS *  DSRED
 603  IGPOYD =  0

C     SPECIAL RETURN FOR NO POINT EVALUATION
 604  RETURN
```



```
C     OUTPUT CONTROL VALUES RE-EVALUATION
 701  ICNTRL =  701
      ENGSVE =  ENERGY
      ENGOUT =  ENERGY  /  1000000.0
      ENERGY =  ENERGY   -   DENRGY
      TENRGY =  ENERGY  *  2.0
      IF( ENERGY -  ENRGTF ) 761, 7611,801
 702  ICNTRL =  702
      FMNOUT =  FMLMNU
      FMLMNU =  FMLMNU   -   DMACHM
      FMLMN2 =  FMLMNU  *  FMLMNU
      IF( FMLMNU -  FMLMNF ) 762, 801, 801
 703  ICNTRL =  703
      VMIN   =  VMIN      -   DVMIN
      VMIN2  =  VMIN  *  VMIN
      IF(  VMIN  -   VMINF ) 763, 801, 801
 704  ICNTRL =  704
      YMAX   =  YMAX      +   DYMIN
      FYFLUT =  - ABS(FYFLUT)
      IF( YMAXF  -    YMAX  ) 901, 801, 801
 705  ICNTRL =  705
      YMIN   =  AMAX1( YMIN-DYMIN, YMINL ) + AMIN1( YMIN-YMINL, 1.0 )
      FYFLUT =  - ABS(FYFLUT)
      IF(  YMIN  -   YMINF ) 765, 801, 801
 706  ICNTRL =  706
      ENDOUT =  ENDMAX
      ENDMAX =  AMIN1( ENDMAX+ENDRPY, ENDREF )
      IF( ENDREF -  ENDOUT ) 901, 901, 801

C     FINAL OUTPUT CONTROL VALUES RE-EVALUATION
 761  ICNTRL =  761
 7611 ENRGYT =  ENERGY   +   ( ABS(DEGYCH)  -  DEGYCH ) * 0.5 * DENRGY
      ENERGY =  AMAX1( ENRGYT, ENRGTL )
      DENRGY =  ABS( DEGYCH )    *    DENRGY
      DEGYCH =  1.0
      ENERGY =  (  ENERGY + SIGN(ENERGY,-ENRGTL+ENRGYT+DENRGY)  ) * 0.5
      TENRGY =  ENERGY  *  2.0
      ENRGTF =  ENERGY
      IF( ENGSVE -  ENERGY ) 901, 901, 801
 762  ICNTRL =  762
      IF( FMLMNU -  FMLMNL ) 901, 801, 801
 763  ICNTRL =  763
      IF(  VMIN  -   VMINL ) 901, 801, 801
 765  ICNTRL =  765
      IF(  YMIN  -   YMINL ) 901, 801, 801

C     SPECIAL RETURN FOR OUTPUT OF INTERMEDIATE OR FINAL VALUES
 801  IGPOYD =  - IGPOYD
      RETURN

C     SPECIAL RETURN FOR OUTPUT OF FINAL VALUES
 901  ILAST  =  1
      GO TO 801

      END
```

13


```
$IBFTC CCK1    FULIST,MAP,DD,DECK
      SUBROUTINE CHECK1

C     SUBROUTINE CHECK1 CODE NUMBER CCECK1                    VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO CHECK THE VALUES FOR THE VEHICLES
C     LONGITUDE, AZIMUTH, LATITUDE, AND FLIGHT PATH ANGLE  -  IF ANY
C     CHECK SHOWS THAT AN INTERMEDIATE OR FINAL OUTPUT IS DESIRED, THE
C     PROPER CONTROL VARIABLES WILL BE SET

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  ODUMP(1),THCNTS,ODUMM(7),THPSTS,DEGATS,ODUMB(2)
      COMMON /CTVAR/   TDUMP(4),GTEMPD,PTEMPD,DTEMPD,TTEMPD,TDUMB(8)
      COMMON /CAVAR/   ADUMP(30),DS,ADUMB(1)
      COMMON /CINTS/   M,N,NTP,IGPOYO,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NOPTEV,IDSTOP,KOUNTF,KOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IF,ITSAVE

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION GTEMPD,PTEMPD,DTEMPD,TTEMPD

C     EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
      EQUIVALENCE        (GTEMP,GTEMPD)
      EQUIVALENCE        (PTEMP,PTEMPD),(DTEMP,DTEMPD),(TTEMP,TTEMPD)

C     COMMON STATEMENTS FOR GOPTHS VARIABLES
      COMMON /CO20/     O2DOUF(25),DSMINF,DSMINT,DSMNGO,ADTEST,ADNEGT,
     1                  AGTEST,AGTSOL,AGSTOP,AGSPOL,AGNEGT,ADGCON
      COMMON /CO40/     PFLOWT,PMAZ,PMAXF,PMAXOL,PMIN,PMINOL,PMINF,DPMRN,
     1                  TFLOWT,TMAX,TMAXF,TMAXOL,TMIN,TMINOL,TMINF,DTMRN
      COMMON /CO100/    O100DF(9),TANDEO,O100UM(84),TPNCOS(3),O101BM(9),
     1                  TANCHK,TANORF,TANCPF

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /COSSFT/  DSDUMP(5),DLTEST,DSRED,DMAX1,DMAX2,DMAXC
      COMMON /CPITCH/  PIDUMP(2),ATKALP,PIDUMB(9)
```



```
C     CHECKS ON THE LONGITUDE AND AZIMUTH
 510  IF( THPSTS ) 511, 514, 516
 511  IF(  TTEMP -  TMAX   ) 512, 711, 561
 512  IF(  TMIN  - TTEMP   ) 5120,712, 562
 5120 IF(     - TFLOWT     ) 5121,514, 5125
 5121 IF( -  TMAX + TTEMP  +  DTMRN  ) 5122,5123,5123
 5122 TMAX    =  TMAX   -   DTMRN
      GO TO 5121
 5123 IF(     TMIN - TTEMP  +  DTMRN  ) 5124,514, 516
 5124 TMIN    =  TMIN   +   DTMRN
      GO TO 5123
 5125 IF( KOUNTL -  KOUNT  ) 5126,514, 514
 5126 IF( IGNORE ) 514, 5127,514
 5127 TFLOWT =  1.0
      GO TO 5121
 514  IF(  PTEMP -  PMAX   ) 515, 714, 564
 515  IF(  PMIN  - PTEMP   ) 5150,715, 565
 5150 IF(     - PFLOWT     ) 5151,516, 5155
 5151 IF( -  PMAX + PTEMP  +  DPMRN  ) 5152,5153,5153
 5152 PMAX    =  PMAX   -   DPMRN
      GO TO 5151
 5153 IF(     PMIN - PTEMP  +  DPMRN  ) 5154,516, 516
 5154 PMIN    =  PMIN   +   DPMRN
      GO TO 5153
 5155 IF( KOUNTL -  KOUNT  ) 5156,516, 516
 5156 IF( IGNORE ) 516, 5157,516
 5157 PFLOWT =  1.0
      GO TO 5151
C     CHECKS ON THE LATITUDE AND  FLIGHT PATH ANGLE
 516  IF( DEGATS ) 517, 518, 519
 517  IF( ADTEST * DTEMP  ) 5170,7170,567
 5170 ADNEGT =  - 1.0
 518  IF( AGTEST *  GTEMP  ) 5180,7180,568
 5180 AGNEGT =  - 1.0
 519  IF( -DMAX2 +  ABS(DTEMP)  ) 520, 569, 569

C     NORMAL RETURN FROM SUBROUTINE CHECK1
 520  RETURN

C     TRANSFER CONTROL FOR CHECKS ON THE TEMPORARY VALUES
 561  NOPTEV =  561
      IF( DSMINF - DS ) 601, 711, 711
 562  NOPTEV =  562
      IF( DSMINF - DS ) 601, 712, 712
 564  NOPTEV =  564
      IF( DSMINF - DS ) 601, 714, 714
 565  NOPTEV =  565
      IF( DSMINF - DS ) 601, 715, 715
 567  NOPTEV =  567
      IF( DSMNGO - DS ) 601, 717, 717
 568  NOPTEV =  568
      IF( DSMNGO - DS ) 601, 718, 718
 569  NOPTEV =  569
      IF( DSMINF - DS ) 609, 719, 719
```



```
C     POINT CHECK CONTROL IMPLEMENTATION

C     NO POINT EVALUATION - OUTPUT CHECK CONTROL NOT PASSED
 601  IJNORE =  1
 602  DSRED  =  0.5
      DS     =  DS *  DSRED
 603  IGPOYO =  0

C     SPECIAL RETURN FOR NO POINT EVALUATION
 604  RETURN

 609  IGNORE =  -1
      DLTEST =  DLTEST * 2.0  -  1.0
      GO TO 602

C     OUTPUT CONTROL VALUES RE-EVALUATION
 711  ICNTRL =  711
      TMAX   =  TMAX     +   DTMRN
      TFLOWT =  - ABS(TFLOWT)
      IF( TMAXF  -   TMAX  ) 901, 801, 801
 712  ICNTRL =  712
      TMIN   =  TMIN     -   DTMRN
      TFLOWT =  - ABS(TFLOWT)
      IF(  TMIN  -   TMINF ) 901, 801, 801
 714  ICNTRL =  714
      PMAX   =  PMAX     +   DPMRN
      PFLOWT =  - ABS(PFLOWT)
      IF( PMAXF  -   PMAX  ) 901, 801, 801
 715  ICNTRL =  715
      PMIN   =  PMIN     -   DPMRN
      PFLOWT =  - ABS(PFLOWT)
      IF(  PMIN  -   PMINF ) 901, 801, 801
```



```
C     OUTPUT CONTROL VALUES RE-EVALUATION
 7170 IF( ADNEGT ) 717, 518, 717
 717  ICNTRL =  717
      ADTEST =  - ADTEST
      ADTSTM =  - ADTEST
      ADNEGT =  0.0
 7176 IF( DTEMP  ) 7177,7179,7177
 7177 IF(  ABS(DTEMP) - ADGCON  ) 7178,801, 801
 7178 DTEMPD =  DSIGN( 0.009, ADTSTM )
      GO TO 801
 7179 DTEMPD =  DSIGN( 0.009, ADTEST )
      GO TO 801
 7180 IF( AGNEGT ) 718, 519, 718
 718  ICNTRL =  718
      AGTSTM =  - AGTEST
      AGTEST =  SIGN( ABS(AGTEST)-1.0, AGTSTM )
      AGTSTM =  - AGTEST
      AGSTOP =  AGSTOP  -  1.0
      AGNEGT =  0.0
 7181 IF( THCNTS ) 7182,7186,7186
 7182 THCNTS =  THCNTS  +  0.5  +  SIGN( 0.5, AGTEST )
      IF( TANCHK -  TANDEO ) 7183,7186,7186
 7183 TANCHK =  AMAX1(  (( AMAX1(   AMIN1(     (TANDEO**0.25 )-TPNCOS(1),
     1                                  AMAX1(  (TANDEO**0.25 )-TPNCOS(2),
     2                TPNCOS(3)  )    1.   0.0   )    )) ** 4,     TANCHK     )
      IGNORE =  0
      ATKALP =  SIGN( 1.0, AGTSTM )
      TANCPF =  1.0E20
      IF( THCNTS ) 7184,7186,7186
 7184 IF( AGTEST ) 7185,7186,7186
 7185 TANCPF =  0.0
 7186 IF( GTEMP  ) 7187,7189,7187
 7187 IF( ABS(GTEMP)  -  ADGCON ) 7188,801, 801
 7188 GTEMPD =  DSIGN( 0.009, AGTSTM )
      IF( AGSTOP ) 801, 901, 801
 7189 GTEMPD =  DSIGN( 0.009, AGTEST )
      IF( AGSTOP ) 801, 901, 801

 719  ICNTRL =  719
      ILAST  =  - 1
      GO TO 801

C     SPECIAL RETURN FOR OUTPUT OF INTERMEDIATE OR FINAL VALUES
 801  IGPOYO =  - IGPOYO
 8011 IF( ICNTRL -  719    ) 8016,8012,8016
 8012 IF( IGNORE           ) 8016,8013,8016
 8013 NSTP   =  NSTP     +   1
 8016 RETURN

C     SPECIAL RETURN FOR OUTPUT OF FINAL VALUES
 901  ILAST  =  1
      GO TO 801

      END
```

14

SUBROUTINE CCECK2 6.0 PAGE 1

```
STOPTC CCK2    FULIST,PMAP,ED,DECK
      SUBROUTINE CHECK2

C     SUBROUTINE CHECK2 CODE NUMBER CCECK2                     VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO CHECK THE VALUES FOR THE VEHICLES
C     ACCELERATION, RADIAL ANGLE, FLIGHT TIME, AND FLIGHT DISTANCE -
C     IF ANY CHECK SHOWS THAT AN INTERMEDIATE OR FINAL OUTPUT IS
C     DESIRED, THE PROPER CONTROL VARIABLES WILL BE SET

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CCNTRL/  DDUMF(12),DTSWTS
      COMMON /CTVAR/   TDUMF(12),TITEMP,TOUMT1,DTEMP,TDUMM(3),STEMPD
      COMMON /CAVAR/   ADUMF(30),DS,ADUMB(1)
      COMMON /CINTS/   N,M,NTP,IGPOTO,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NOPTEV,IOSTOP,KOUNTP,KOUNTL,NSTP,NUSL,
     2                 NSTOTT,NUSLTT,IT,ITSAVE

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION STEMPD

C     COMMON STATEMENTS FOR GOPTHS VARIABLES
      COMMON /CO19/    ACRLTS,AFLOUT,ACEXTS,ACCEL,ACCEL2,AMAX,AMAX2,
     1                 AMAXP,AMAXOL,AMIN,AMIN2,AMINOL,AMINP,DAMIN,
     2                 TESTA2,TTA2OL,ACCLOS(9),AMOLD2,ALOLD2,ACC2OM,
     3                 ACC2OL,ACC2MX,ACC2PP,OLDDUD(12)
      COMMON /CO20/    TIMAX,TIMAXP,TIMAXL,TIMEOL,DTIMEN,BMAX,BMAXP,
     1                 BMAXOL,DDMEQ,SAMOUT,SDMAX,SDMAXP,SDMAXL,SDMXOL,
     2                 SNTEDD,DSSAVE,DSMAXS,DSPAT,DSMIN,DSMEAD,DSMINP,
     3                 M2DDUD(10)

C     DOUBLE PRECISION STATEMENTS FOR GOPTHS VARIABLES
      DOUBLE PRECISION SDMAX,SDMAXP,SDMAXL,SDMXOL,SNTEDD

C     EQUIVALENCE STATEMENTS FOR GOPTHS VARIABLES
      EQUIVALENCE        (IAC2PP,ACC2PP)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CMSSET/  DSDUMF(6),DSPED,DSDUMB(3)
      COMMON /COCTAL/  OM400,OO75,OO3774,OO777
```

SUBROUTINE CCECK2 6.0 PAGE 2

```
C     CHECKS ON THE ACCELERATION
 520  IF( ACRLTS ) 521, 523, 526
 521  IF( ACCEL2 -  AMAX2  ) 522, 721, 571
 522  IF( AMIN2  -  ACCEL2 ) 5220,722, 572
 5220 IF(     - AFLOUT       ) 5221,523, 5225
 5221 IF( - ( AMAX-DAMIN )**2   +  ACCEL2 ) 5222,5223,5223
 5222 AMAX   =  AMAX   -   DAMIN
      AMAX2  =  AMAX  *  AMAX
      GO TO 5221
 5223 IF(   ( AMIN+DAMIN )**2   -   ACCEL2 ) 5224,523, 523
 5224 AMIN   =  AMIN   +   DAMIN
      AMIN2  =  AMIN  *  AMIN
      GO TO 5223
 5225 IF( KOUNTL -  KOUNT   ) 5226,523, 523
 5226 IF( IGNORE ) 523, 5227,523
 5227 AFLOUT =  1.0
      GO TO 5221
 523  IF(-TESTA2 ) 524, 526, 525
 524  ACC2MP =  OR(  AND( ACCEL2, OO777  ),  ACC2OM  )
      IF( ACC2PP -  ACCEL2 +  AMOLD2 ) 526, 724, 574
 525  ACC2PP =  OR(  AND( ACCEL2, OO777  ),  ACC2OL  )
      IF( ACC2PP +  ACCEL2 -  ALOLD2 ) 526, 725, 575
C     CHECKS ON THE TIME, RADIAL ANGLE, AND PATH LENGTH
 526  IF( DTSWTS ) 527, 528, 529
 527  IF( DTEMP  -   BMAX   ) 528, 727, 577
 528  IF( TITEMP -  TIMAX   ) 529, 728, 578
 529  IF( STEMPD -  SDMAX   ) 530, 729, 579

C     NORMAL RETURN FROM SUBROUTINE CHECK2
 530  RETURN

C     TRANSFER CONTROL FOR CHECKS ON THE TEMPORARY VALUES
 571  NOPTEV =  571
      IF( DSMINP - DS ) 601, 721, 721
 572  NOPTEV =  572
      IF( DSMINP - DS ) 601, 722, 722
 574  NOPTEV =  574
      ACC2PM =  0.0
 5744 IF(   - ACEXTS    ) 5748,5749,5748
 5748 IF(ACC2MX-ACCEL2) 5749,7748,7748
 5749 IF( DSMINP - DS ) 601, 724, 724
 575  NOPTEV =  575
      ACC2OL =  0.0
 5756 IF(   ACEXTS      ) 7759,7758,7758
 5759 IF( DSMINP - DS ) 601, 725, 725
 577  NOPTEV =  577
      IF( DSMINP - DS ) 601, 727, 727
 578  NOPTEV =  578
      IF( DSMINP - DS ) 601, 728, 728
 579  NOPTEV =  579
      IF( DSMINP - DS ) 601, 729, 729
```

SUBROUTINE CCECK2 6.0 PAGE 3

```
C     POINT CHECK CONTROL IMPLEMENTATION

C     NO POINT EVALUATION - OUTPUT CHECK CONTROL NOT PASSED
 601  IGNORE =  1
 602  DSPED  =  0.5
      DS     =  DS *  DSPED
 603  IGPOTO =  0

C     SPECIAL RETURN FOR NO POINT EVALUATION
 604  RETURN

C     OUTPUT CONTROL VALUES RE-EVALUATION
 721  ICNTRL =  721
      AMAX   =  AMAX     +  DAMIN
      AMAX2  =  AMAX  *  AMAX
      AFLOUT =  - ABS(AFLOUT)
      IF( AMAXP  -   AMAX  ) 901, 801, 801
 722  ICNTRL =  722
      AMIN   =  AMIN     -  DAMIN
      AMIN2  =  AMIN  *  AMIN
      AFLOUT =  - ABS(AFLOUT)
      IF(  AMIN  -   AMINP ) 901, 801, 801
 724  ACC2PP =  OR(  AND( ACCEL2, OO777  ),  ACCLOS(6)   )
      ACC2PP =  ACC2PP  -  ACCEL2  +  AMOLD2
      ICNTRL =  724 + ISIGN( 50, NSTP-1 ) - 50 + ISIGN( 5, IAC2PP ) + 5
      ACC2PP =  OR(  AND( ACCEL2, OO777  ),  ACCLOS(7)   )
      ACC2PP =  ACC2PP  -  ACCEL2  +  AMOLD2
      ICNTRL =  ICNTRL - ( ISIGN( 1, IAC2PP ) + 1 ) / 2
      AMOLD2 =  - AMOLD2
      ACC2OM =  ACCLOS(4)
      TESTA2 =  - 1.0
      ACC2MX =  AMAX1( ( AMAX1(  AMIN1(  SQRT(ACCEL2)-ABS(ACCLOS(1)),
     1                             AMAX1( SQRT(ACCEL2)-ABS(ACCLOS(2)),
     2              ABS(ACCLOS(3)) )  1,  0.0  )   )) ** 2,      ACC2MX    )
 7746 IF(-ACEXTS ) 7747,801, 801
 7747 IF( ACCEL2 -  ACC2MX ) 526, 526, 774
 7748 TESTA2 =  - 1.0
      AMOLD2 =  - AMOLD2
      ACC2OM =  ACCLOS(4)
      GO TO 526
 725  ACC2PP =  OR(  AND( ACCEL2, OO777  ),  ACCLOS(8)   )
      ACC2PP =  ACC2PP  +  ACCEL2  -  ALOLD2
      ICNTRL =  725 + ISIGN( 50, NSTP-1 ) - 50 + ISIGN( 5, IAC2PP ) + 5
      ACC2PP =  OR(  AND( ACCEL2, OO777  ),  ACCLOS(9)   )
      ACC2PP =  ACC2PP  +  ACCEL2  -  ALOLD2
      ICNTRL =  ICNTRL + ( ISIGN( 1, IAC2PP ) + 1 ) / 2
      ALOLD2 =  1.0E6
      ACC2OL =  ACCLOS(5)
      TESTA2 =     1.0
 7756 IF(-ACEXTS ) 526, 526, 801
 7758 TESTA2 =     1.0
      ALOLD2 =  1.0E6
      ACC2OL =  ACCLOS(4)
      GO TO 526
```

SUBROUTINE CCECK2 6.0 PAGE 4

```
C     OUTPUT CONTROL VALUES RE-EVALUATION
 727  ICNTRL =  727
      BMAX   =  BMAX      +  DAMIN
      IF( BMAXP  -   BMAX  ) 901, 801, 801
 728  ICNTRL =  728
      TIMAX  =  TIMAX     +  DTIMEN
      IF( TIMAXP -   TIMAX ) 750, 801, 801
 729  ICNTRL =  729
      SAMOUT =  SDMAX  /  1000000.000
      SDMAX  =  SDMAX   +    SNTEDD
      DSSAVE =  DSMAXS
      IOSTOP =  - 1
      IF( SDMAXP -  SDMAX  ) 759, 801, 801
 759  ICNTRL =  730
      SAMOUT =  SDMAX  /  1000000.000
      SDMAX  =  SDMAX   +    SNTEDD
      DSSAVE =  DSMAXS
      IOSTOP =  - 1
      IF( SDMAXP -  SDMAX  ) 760, 801, 801

C     FINAL OUTPUT CONTROL VALUES RE-EVALUATION
 750  ICNTRL =  750
      IF( TIMAXL -   TIMAX  ) 901, 801, 801
 759  ICNTRL =  759
      IF( SDMAXL -   SDMAX  ) 901, 801, 801
 760  ICNTRL =  760
      IF( SDMAXL -   SDMAX  ) 901, 801, 801

C     OUTPUT CONTROL MAXIMUM VALUES RE-EVALUATION
 774  ACC2MX =  AMAX1( ( AMAX1(   AMIN1(  SQRT(ACCEL2)-    ACCLOS(1),
     1                              AMAX1( SQRT(ACCEL2)-    ACCLOS(2)),
     2             ABS(ACCLOS(3)) )  1,  0.0  )   )) ** 2,      ACC2MX     )
      GO TO 851

C     SPECIAL RETURN FOR OUTPUT OF INTERMEDIATE OF FINAL VALUES
 801  IGPOTO =     IGPOTO
      RETURN

C     SPECIAL RETURN FOR OUTPUT OF FINAL VALUES
 901  ILAST  =  1
      GO TO 801

      END
```

15


```
$IBFTC CCKT    FULIST,PHAP,DD,DECK
      SUBROUTINE CHECKT

C     SUBROUTINE CHECKT CODE NUMBER CCECKT                    VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO CHECK THE VARIOUS VALUES FOR THE
C     VEHICLES SKIN TEMPERATURE  -  IF ANY CHECK SHOWS THAT AN
C     INTERMEDIATE OR FINAL OUTPUT IS DESIRED, THE PROPER CONTROL
C     VARIABLES WILL BE SET

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  DDUMF(1),THCNTS,DDUMB(1)
      COMMON /CVAR/    VDUMF(16),TIME,VDUMM(9),TEMPS(2,6),VDUMB(2)
      COMMON /CTVAR/   TDUMF(12),TITEMP,TOUNTI,TDUMB(6)
      COMMON /CAVAR/   ADUMF(18),ATPS(2,6),DS,ADUMB(1)
      COMMON /CINTS/   M,N,NTP,IGPOYD,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NOPTEV,IOSTOP,KOUNTF,KOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IT,ITSAVE

C     COMMON STATEMENTS FOR GOPINS VARIABLES
      COMMON /CO20/    O2UDUMF(22),DSMAX,DSMIN,DSMINO,DSMINF,DSMINT,
     1                 O2DOUB(9)
      COMMON /CO100/   TEMPTS,TPERTS,THEXTS,TCEXTS,TPNDEG,TANDEG,TPEGRT,
     1                 TPMAX(6),ATPHTT(6),ATPLTT(6),P1(6),P2(6),PO(6),
     2                 P1CNST,P2CNST,POCNST,ATRADS(6),CNOS(6),E(6),
     3                 O1OODM(12),TFSTEU,TTCOOL,TEMPBS(3),TPNBOS(9),
     4                 TPNCAS(3),TANNOL,TALNOL,TANGAM,TANBDL,TANBRK,
     4                 TANCHE,TANORF,TANCAP

C     DIMENSION STATEMENTS FOR GOPINS VARIABLES
      DIMENSION        ITNOAS(9)

C     EQUIVALENCE STATEMENTS FOR GOPINS VARIABLES
      EQUIVALENCE      (ITNOSS,TPNOAS),(ITNORF,TANORF)
      EQUIVALENCE      (ITNGAM,TANGAM),(ITNOBL,TANOBL)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CDSSET/  DSDUMF(4),DSRED,DSDUMB(3)
      COMMON /COCTAL/  OM409,O7T5,O83774,O8777
      COMMON /CPITCH/  ATTACK,ATAKNF,ATKALP,PIDUMB(19)

C     LOCAL DATA VALUES
      DATA   TISAVE,TINTS1,TIBIS2,TIBIO1/0.0,1.0,2.0,60.0/
```



```
C     CHECKS ON THE TEMPERATURES
 539  IF( TEMPTS ) 531, 534, 541
 531  IF(  NTP   ) 5311,536, 5311
 5311 IF(                     TISAVE ) 5312,532, 5312
 5312 IF( TIME    -  TIBIS1 -  TISAVE ) 5313,5314,5314
 5313 IF( TITEMP -  TIBIS2 -  TISAVE ) 536, 537, 532
 5314 TISAVE  =  0.0
 532  IF( TESTED ) 533, 534, 533
 533  DO 5330 IT = 1, NTP
      IF( SIGN( ATPHTT(IT), ATPS(1,IT) )  ) 583, 5330,5330
 5330 CONTINUE
      GO TO 537
 534  DO 5340 IT = 1, NTP
      IF( SIGN( ATPLTT(IT), ATPS(1,IT) )  ) 5340,5340,584
 5340 CONTINUE
      GO TO 537
 536  IF(   MOD( NTP  )    ) 537, 541, 537
 537  IF(-TESTED ) 541, 538, 539
 538  TANOAF  =  OR(  AND( TANDEG, O8777  ),  TANGAM  )
      IF( TANOAF -  TANDEG +  TANOML ) 541, 738, 588
 539  TANOAF  =  OR(  AND( TANDEG, O8777  ),  TANOBL  )
      IF( TANOAF +  TANDEG -  TALNOL ) 541, 739, 589
 540  IF( TANDEG -  TANCHE -  TANCAP ) 541, 740, 590

C     NORMAL RETURN FROM SUBROUTINE CHECKT
 541  RETURN

C     SPECIAL RETURN WHEN NO OUTPUT IS DESIRED
 5411 TISAVE  =  TITEMP
      IGNORE  =  0
      NSTP    =  ( NSTP / 100 ) * 10    +   MOD( NSTP, 100 )    +    1
      NSTPTT  =  NSTPTT  +   NSTP
      NUSLTT  =  NUSLTT  +   NUSL
      NSTP    =  0
      NUSL    =  0
      RETURN
```



```
C     TRANSFER CONTROL FOR CHECKS ON THE TEMPORARY VALUES
 583  NOPTEV =  583
      IF(  - TPERTS    ) 5838,5839,5839
 5838 IF( TPMAX(IT) - TEMPS(1,IT)     ) 5839,7338,7338
 5839 IF( DSMINT - DS ) 600, 733, 733
 584  NOPTEV =  584
 5848 IF(    TPERTS     ) 5849,7348,7348
 5849 IF( DSMINT - DS ) 600, 734, 734
 588  NOPTEV =  588
      IF( ATTACK ) 5882,5881,5882
 5881 TANGAM =  0.0
      DSMINE =  DSMINF
      GO TO 5883
 5882 DSMINE =  DSMINT
 5883 IF(  - THCNTS    ) 5884,5886,5886
 5884 IF(TANDEG-TANCHE) 5886,5886,5889
 5886 IF(  - THEXTS    ) 5888,5889,5889
 5888 IF(TANGAM-TANDEG) 5889,7388,7388
 5889 IF( DSMINE - DS ) 601, 738, 738
 589  NOPTEV =  589
      IF( ATTACK ) 5892,5891,5892
 5891 TANOBL =  0.0
      DSMINE =  DSMINF
      GO TO 5893
 5892 DSMINE =  DSMINT
 5893 IF(  - THCNTS    ) 5894,5896,5896
 5894 IF(TANDEG-TANCHE) 5896,5896,5899
 5896 IF(    THEXTS     ) 5899,7398,7398
 5899 IF( DSMINE - DS ) 601, 739, 739
 590  NOPTEV =  590
 5908 IF(  - THCNTS    ) 5909,7408,5909
 5909 IF( DSMINT - DS ) 604, 740, 740
```



```
C     POINT CHECK CONTROL IMPLEMENTATION

C     NO POINT EVALUATION - OUTPUT CHECK CONTROL NOT PASSED
 601  IGNORE =  1
 602  DSRED  =  0.9
      DS     =  DS *  DSRED
 603  IGPOYD =  0

C     SPECIAL RETURN FOR NO POINT EVALUATION
 604  RETURN

 606  IF( IGNORE ) 602, 6061,607
 6061 IGNORE =  IGNORE  +  ISIGN( 2, IGNORE )
      GO TO 602

 608  IGNORE =  -1
 6081 IF(       NSTP          - 15 ) 6082,6083,6083
 6082 DSRED  =  0.125
      DS     =  DS    *   DSRED
      GO TO 603
 6083 DSRED  =  0.25
      DS     =  DS    *   DSRED
 6084 IF( MOD( NSTP, 100 ) - 15 ) 603, 6085,6085
 6085 NSTP   =  NSTP     +    50
      GO TO 6084

C     OUTPUT CONTROL VALUES RE-EVALUATION
 733  ICNTRL =  900    +    IT
      ITSAVE =  IT
      TPMAX(IT)  =  AMAX1(   AMIN1(  (TEMPS(1,IT)+ATPS(1,IT))-TEMPS(1),
     1                       AMAX1(  (TEMPS(1,IT)+ATPS(1,IT))-TEMPS(2),
     2                             ABS(TEMPS(3))  )  ),   TPMAX(IT)    )
      ATPHTT(IT)  =  0.0
      ATPLTT(IT)  =  1.0
      GO TO 591
 7338 ATPHTT(IT)  =  0.0
      ATPLTT(IT)  =  1.0
      GO TO 537
 734  ICNTRL =  950    +    IT
      ITSAVE =  IT
      ATPHTT(IT)  =  1.0
      ATPLTT(IT)  =  0.0
      GO TO 601
 7348 ATPHTT(IT)  =  1.0
      ATPLTT(IT)  =  0.0
      GO TO 537
```

```
C     OUTPUT CONTROL VALUES RE-EVALUATION
 730  ICNTRL =  730
      TPMDEO =  TANDEO ** 0.25
      TANOMX =  AMAX1( (( AMAX1(   AMIN1(   TPMDEO - ABS( TPMOBS(1))  ),
     1                                   AMAX1(  TPMDEO - ABS( TPMOBS(2)  ),
     2            ABS(TPMOBS(3)) )  ),  0.0  )    )) ** 4,     TANOMX      )
      TANMSV =  TANMOL
      TANMOL =  - TANMOL
      TANOBM =  TPMOBS(4)
      TESTPO =  - 1.0
 7381 IF(-TNCNTS ) 7382,7386,7386
 7382 TNCNTS =  TNCNTS  -  1.0
      IF( TANCNX -  TANDEO ) 7383,7386,7386
 7383 TANCNX =  AMAX1(  (( AMAX1(   AMIN1(   TPMDEO -        TPMCOS(1)),
     1                                   AMAX1(  TPMDEO -        TPMCOS(2)),
     2            ABS(TPMCOS(3)) )  ),  0.0  )    )) ** 4,     TANCNX      )
      ATKALP =  - 1.0
      TANCRP =  1.0E20
      IF(-TNCNTS ) 7384,7385,7385
 7384 TANCRP =  0.0
 7385 IF(-TCEXTS ) 7386,780, 780
 7386 IF(-TNEXTS ) 7387,780, 780
 7387 IF( TANDEO -  TANOMX ) 5411,5411,7800
 7388 TESTPO =  - 1.0
      TANMOL =  - TANMOL
      TANOBM =  TPMOBS(4)
      IF(-TNCNTS ) 7389,541, 541
 7389 TNCNTS =  TNCNTS  -  1.0
      TANCRP =  0.0
      IF(-TNCNTS ) 541, 7400,7400
 739  ICNTRL =  739
      TALMSV =  TALMOL
      TALMOL =  1.0E20
      TANOBL =  TPMOBS(5)
      TESTPO =  0.0
 7391 IF(-TNCNTS ) 7392,7396,7396
 7392 IF( TANCNX -  TANDEO ) 7393,7396,7396
 7393 ATKALP =  1.0
      TANCRP =  1.0E20
 7395 IF(-TCEXTS ) 7396,7396,789
 7396 IF(-TNEXTS ) 7890,5411,789
 7390 TESTPO =  0.0
      TALMOL =  1.0E20
      TANOBL =  TPMOBS(5)
      GO TO 541
 740  ICNTRL =  740
      IGNORE =  MOD( IGNORE, 2 )
      ATKALP =  1.0
      TANCRP =  1.0E20
      IF(-TCEXTS ) 5412,5411,801
 7450 TANCRP =  1.0E20
      GO TO 541
```



```
C     OUTPUT CONTROL MAXIMUM VALUES RE-EVALUATION
 7803 TINOMX =  TITEMP
      TANOMX =  AMAX1(  (( AMAX1(   AMIN1(   TPMDEO -        TPMOBS(1)),
     1                                   AMAX1(  TPMDEO -        TPMOBS(2)),
     2            ABS(TPMOBS(3)) )  ),  0.0  )    )) ** 4,     TANOMX      )
 780  IF( ATTACK ) 7802,7801,7802
 7801 ICNTRL =  780
 7802 TANORP =  AND( TANDEO, O8777 )
      ITNORP =  ITNOBM  +   ITNOBS(6)  -   ITNORP
      IF( TANDEO  -  TANORP  .GT.  TANMSV ) GO TO 801
      ICNTRL =  ICNTRL  -   1
      ITNORP =  ITNORP  +   ITNOBS(6)
      IF( TANDEO  -  TANORP  .GT.  TANMSV ) GO TO 801
      ICNTRL =  ICNTRL  -   1
      ITNORP =  ITNORP  +   ITNOBS(7)
      IF( TANDEO  -  TANORP  .GT.  TANMSV ) GO TO 801
      ICNTRL =  ICNTRL  -   1
      GO TO 801
 7890 IF( TCEXTS .LT.  2.0 ) GO TO 5411
      IF( TITEMP-TIBIAL .LT.  TINOMX ) GO TO 5411
      TANOMX =  AMAX1(   AMAX1( - TPMOBS(1), - TPMOBS(2), - TPMOBS(3),
     1                          0.0 ) ** 4,   TANDEO   )
      IF( TANDEO.LT.TANOMX ) GO TO 5411
      TINOMX =  TITEMP
      TANOMX =  AMAX1(  (( AMAX1(   AMIN1(   TANDEO**0.25- TPMOBS(1)),
     1                                   AMAX1(  TANDEO**0.25- TPMOBS(2)),
     2            ABS(TPMOBS(3)) )  ),  0.0  )    )) ** 4,     TANOMX      )
 789  IF( ATTACK ) 7892,7891,7892
 7891 ICNTRL =  749
 7892 TANORP =  AND( TANDEO, O8777 )
      ITNORP =  ITNOBL  +   ITNOBS(6)  -   ITNORP
      IF( TANDEO  +  TANORP  .LT.  TALMSV ) GO TO 801
      ICNTRL =  ICNTRL  +   1
      ITNORP =  ITNORP  +   ITNOBS(6)
      IF( TANDEO  +  TANORP  .LT.  TALMSV ) GO TO 801
      ICNTRL =  ICNTRL  +   1
      ITNORP =  ITNORP  +   ITNOBS(7)
      IF( TANDEO  +  TANORP  .LT.  TALMSV ) GO TO 801
      ICNTRL =  ICNTRL  +   1
      GO TO 801

C     SPECIAL RETURN FOR OUTPUT OF INTERMEDIATE OR FINAL VALUES
 801  IGPDYD =  - IGPDYD
 8011 TISSVE =  TITEMP
 8312 IGNORE =  0
 8313 NSTP   =  ( NSTP / 100 ) * 10   +   MOD( NSTP, 100 )   +   1
 8314 RETURN

      END
```

```
$IBFTC CCSS     FULIST,FMAP,DECK
      BLOCK DATA

C     SUBROUTINE CCSS0A CODE NUMBER CCSS0A                    VERSION 100.6

C     THIS PROGRAM WILL INCREASE THE SPACE RESERVED FOR LOCAL VARIABLES
C     WHICH ARE IN COMMON STATEMENTS IN SUBROUTINE OUTPUT

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CNSS/     NUPTS,NSTOP

C     DATA STATEMENTS FOR GLOBAL VARIABLES
      DATA   NSTOP/100/

C     COMMON STATEMENTS FOR LOCAL VARIABLES
      COMMON /CSNTRL/  ISNTRL(101)/CSKONT/ISKONT(101)/CSNSTP/ISNSTP(101)
      COMMON /CSNUSL/  ISNUSL(101)/CSS/SS(101)/CSB/SB(101)/CSH/SH(101)
      COMMON /CSA/     SA(101)/CSTNDB/STNDB(101)/CSTP/STP(6,101)

C     TYPE STATEMENTS FOR LOCAL VARIABLES
      COMPLEX          SS


      END
```



```
$IBFTC ACRS     FULIST,FMAP,DD,DECK
      SUBROUTINE AFRONS( SINARG )

C     SUBROUTINE AFRONS CODE NUMBER ACRONS                 VERSION 6.0

C     THIS PROGRAM WILL SEPARATE THE INTEGER AND FRACTIONAL PARTS OF A
C     SINGLE PRECISION NUMBER.
C     NOTE  -  ALL CALLS TO THIS SUBROUTINE ARE ACTUALLY MADE AS
C     FUNCTION SUBPROGRAM CALLS  -  THIS IS THE REASON FOR
C     DOUT    =  DOUT   (WHICH RETURNS THE ARGUMENT IN THE ACCUMULATOR)
C     JUST BEFORE THE SINGLE RETURN STATEMENT.

      DOUBLE PRECISION DOUT
      REAL             SOUT(2)
      EQUIVALENCE      ((OUT1,OUT1,DOUT,SOUT),(IOUT2,SOUT(2)))

C     LOCAL DATA VALUES
      DATA             IBLANK,IERROR/6H       ,6HERROR/
      DATA             IP0,IP2/6H,       ,6H.00000/

      ENTRY      AFR6NS( SINARG )
      BIAS    =  0.0000005
      GO TO 111
      ENTRY      AFR5NS( SINARG )
      BIAS    =  0.000005
      GO TO 111
      ENTRY      AFR4NS( SINARG )
      BIAS    =  0.00005
      GO TO 111
      ENTRY      AFR3NS( SINARG )
      BIAS    =  0.0005
      GO TO 111
      ENTRY      AFR2NS( SINARG )
      BIAS    =  0.005
      GO TO 111
      ENTRY      AFR1NS( SINARG )
      BIAS    =  0.05
      GO TO 111
      ENTRY      AFR0NS( SINARG )
      BIAS    =  0.5
      GO TO 111
```



```
C     ACTUAL STARTING POINT AFTER BIAS HAS BEEN CORRECTLY SET
  111 SIN     =  SINARG
      OUT1    =  AINT( SIN )
      FRACT   =  ABS( SIN - OUT1 )
      IF( FRACT ) 11, 1, 11
    1 IOUT2   =  IBLANK
    2 DOUT    =  DOUT
      RETURN

   11 SIN     =  SIN    +   SIGN( BIAS, SIN )
      OUT1    =  AINT( SIN )
      FRACT   =  ABS( SIN - OUT1 )
   12 IOUT2   =  FRACT * 1000000.0
      IO21    =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IO22    =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IO23    =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IO24    =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IO25    =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IO26    =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IF( IOUT2 ) 15, 13, 15
   13 IOUT2   =  ((((IO26*64+IO25)*64+IO24)*64+IO23)*64+IO22)*64+IO21
      GO TO 2
   15 IOUT2   =  IERROR
      GO TO 2

*     THIS DUMMY ENTRY POINT ALLOWS FOR FUTURE USE OF IFRONA.
      ENTRY AFRONA( BINAGA )
      DOUBLE PRECISION BINAGA
      DOUT    =  BINAGA
      GO TO 2

C     THIS ENTRY POINT WILL INTEGERIZE THE INTEGER PART (FORMERLY
C     EXPRESSED AS A REAL NUMBER) OF THE ARGUMENT AS WELL AS PLACING A
C     DECIMAL POINT IN FRONT OF THE FRACTIONAL PART OF THE ARGUMENT
C     WHICH THIS SECTION ASSUMES WAS CALCULATED FROM AFR-N-NS.
      ENTRY IFRONA( BINAGI )
      DOUBLE PRECISION BINAGI
   21 DOUT    =  BINAGI
      IOUT1   =  OUT1
   22 IF( IOUT2 - IBLANK ) 23, 24, 23
   23 IOUT2   =  IP2    +   (ABS( IOUT2 / 64 )
      GO TO 2
   24 IOUT2   =  IP0
      GO TO 2

      END
```



```
$IBFTC ACRO     FULIST,FMAP,DD,DECK
      DOUBLE PRECISION FUNCTION AFRONO( BINARG )

C     FUNCTION   AFRONO CODE NUMBER ACRONO                VERSION 6.0

C     THIS PROGRAM WILL SEPARATE THE INTEGER AND FRACTIONAL PARTS OF A
C     DOUBLE PRECISION NUMBER WHERE ABS(BINARG) .LE. 34359738368.0.

      DOUBLE PRECISION BINARG,BIN,DOUT,OUT1
      REAL             SOUT(2)
      EQUIVALENCE      ((BIN,BIN),(IOUT1,DOUT,SOUT),(IOUT2,SOUT(2)))

C     LOCAL DATA VALUES
      DATA             IOVFL,IP2,IP0,IER/6HOVERFLW,6H.00000,1H.,6H ERROR/

      BIN     =  BINARG
      IF( DABS(BIN) - 34359738368.0 ) 11, 1, 1
    1 IOUT1   =  ISIGN( 0, IBIN )
      IOUT2   =  IOVFL
    2 AFRONO  =  DOUT
      RETURN
   11 OUT1    =  DINT( BIN )
      IOUT1   =  OUT1
      OUT2    =  BIN   -   OUT1
      IF(  OUT2  )  12,  14,  12
   12 IOUT2   =  AMIN1( ABS(OUT2*100000.0) + 0.5, 99999.0 )
      IOUT21  =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IOUT22  =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IOUT23  =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IOUT24  =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IOUT25  =  MOD( IOUT2,  10 )
      IOUT2   =        IOUT2 / 10
      IF( IOUT2 )  15,  13,  15
   13 IOUT2   = ((((IOUT25*64+IOUT24)*64+IOUT23)*64+IOUT22)*64+IOUT21)+IP2
      GO TO 2
   14 IOUT2   =  IP0
      GO TO 2
   15 IOUT2   =  IER
      GO TO 2


      END
```

11


```
$IBFTC OCTK     PULIST,PMAP,DD,DECK
      SUBROUTINE OUTACK

C     SUBROUTINE OUTACK CODE NUMBER OCTACK                     VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO OUTPUT THE LAST INTERMEDIATE SET OF
C     VALUES AND THE VALUES WHICH WERE STORED FOR LATER OUTPUT WHEN A
C     PREMATURE STOP OCCURS DURING THE PROGRAM

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  SNSSWT,CDUMB(12)
      COMMON /COTCTL/  IOTCTL(15)
      COMMON /CVAR/    V2,VDUMY2,VDUMM(2),RD,VDUMB(20)
      COMMON /CPVAR/   PDUMP(2),FV,PDUMM(3),FY,PDUMB(19)
      COMMON /CINTS/   N,N,NTP,ISPDYD,IGNORE,ICOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NOPTEV,IDSTOP,ICOUNTF,ICOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IT,ITSAVE

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION RD

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CRADUS/  RSHAPZ,RADUMM(21),RZEROD,RADUMB(2)

C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION RZEROD

C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE      (AA,OUT)
```



```
 401  ICNTRL =  700
      JFAIL  =  JFAIL   +   1
      IOTCTL(JFAIL)  =  ICNTRL
 403  IF(   DBLE(RSHAPE)    ) 404, 405, 404
 404  IRPOTD =  100
      CALL PUNCLU
      GO TO 406
 405  FV     =  RD     -     RZEROD
      FV     =  SQRT(V2)
 406  CONTINUE

C     OUTACK CONTROL SENSE SWITCH CONTROL
 470  IF(    SNSSWT     ) 4701,4703,4701
 4701 IF( SWITCH(47C) ) 4702,4703,4702
 4702 AA = OUT
 4703 CONTINUE

 471  CALL OUTPUT
 472  CALL OUTPNL
 473  RETURN
```

END


```
$IBFTC OCTP     PULIST,PMAP,DD,DECK
      SUBROUTINE OUTPUT

C     SUBROUTINE OUTPUT CODE NUMBER OCTPUT                     VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO OUTPUT THE INTERMEDIATE VALUES OF THE
C     CALCULATION AND TO STORE THOSE VALUES WHICH ARE NOT IMMEDIATELY
C     OUTPUT ONTO THE OUTPUT TAPE

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CNSS/    NNPPTS,NSTOP
      COMMON /CONTRL/  SNSSWT,THCNTS,GOPINS,NOCASE,CNTMAX,THE,SNATCT,
     1                 CDUMB(6)
      COMMON /COTCTL/  IOTCTL(15)
      COMMON /CFORMT/  FB3(25),FB9(25)
      COMMON /CSAR/    V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,BETAD,
     1                 ENDURD,TEMPD,SD
      COMMON /CPVAR/   PDUMP(2),FV,PDUMM(3),FY,PDUMB(19)
      COMMON /CSCAR/   ALPHAD,SADUMM(3),PHIDES,PHIOLD,SINPHI,COSPHI
      COMMON /CINTS/   N,N,NTP,ISPDYD,IGNORE,ICOUNT,INTLST,ILAST,JFAIL,
     1                 ICNTRL,NOPTEV,IDSTOP,ICOUNTF,ICOUNTL,NSTP,NUSL,
     2                 NSTPTT,NUSLTT,IT,ITSAVE
      COMMON /CPIES/   PDUMP(19),RDDRGR,BDDRDR,PDUMB(6)

C     TYPE STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE  PRECISION V2D,VD,RD,GAMMAD,PSID,DELTAD,THETAD,TIMED,BETAD
      DOUBLE PRECISION ENDURD,TEMPD(6),SD
      REAL              TEMPS(2,6)

C     EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
      EQUIVALENCE      (V2,V2D),(V,VD),(R,RD),(GAMMA,GAMMAD),(PSI,PSID)
      EQUIVALENCE      (DELTA,DELTAD),(THETA,THETAD),(TIME,TIMED)
      EQUIVALENCE      (BETA,BETAD),(ENDURE,ENDURD),(TEMPS,TEMPD),(S,SD)

C     COMMON STATEMENTS FOR GOPINS VARIABLES
      COMMON /C01/     O1DUMP(6),FMACHN,FMACH2,FMNOUT,O1DUMB(6)
      COMMON /C02/     O2DUMP(3),ENGOUT,O2DUMB(8),ENGCON
      COMMON /C0119/   O1GDUP(3),ACCEL,ACCEL2,O1GDUM(26),ENDOUT,OT101(1)
      COMMON /C020/    O20DUP(9),SMNOUT,O20DUB(26)
      COMMON /C01190/  O1090P(6),TPNDEG,TANDEG,TPEGNT,O1090M(33),PB(6),
     1                 P1CNST,P2CNST,PBCNST,PTPABS(6),CNOS(6),E(6),
     2                 O1090B(37)
```



```
C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CUSSET/  ANGINC,DSDUMB(9)
      COMMON /CPTTS/   THENTY,PTDUMB(12)
      COMMON /CRANGE/  GITVCT,DRUMM,CROSSR,PSIRG,PSIRGA
      COMMON /CPITCH/  PIDUMP(21),DIASPD

C     COMMON STATEMENTS FOR LOCAL VARIABLES
      COMMON /CSNTPL/  ISNTPL(50)/CSHBNT/ISHDNT(50)/CSNSTP/ISNSTP(50)
      COMMON /CSNUSL/  ISNUSL(50)/CSS/SS(50)/CSD/SD(50)/CSA/SA(50)
      COMMON /CSA/     SA(50)/CSTHDO/STHDO(50)/CSTP/STP(6,50)

C     MISCELLANEOUS TYPE AND DIMENSION STATEMENTS FOR LOCAL VARIABLES
      COMPLEX          FVOUT,FYOUT,GAMOUT,PSIOUT,DELOUT,THEOUT,TIMOUT
      COMPLEX          ALPOUT,PHIOUT,ACCOUT,TOPOUT,SMNOUT,CASOUT,PSNOUT
      COMPLEX          SS,AFPOMD,AFPOMA,ITEOMA
      COMPLEX          AFRBMS,AFPBMS,AFP9MS,AFP4MS,AFP3MS,AFP2MS,AFP1MS
      EXTERNAL         AFPOMS,AFPOMS,AFP9MS,AFP4MS,AFP3MS,AFP2MS,AFP1MS
      DIMENSION        TPHAE(6),ITPHSE(6),TPEOM(6)

C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE      (AA,OUT)

C     LOCAL DATA VALUES
      DATA   ONTCSE/0/
      DATA   SEJECT,SBLANK,TPHEAD/1H1,6H      ,6H  TEMP/

C     FORMATS
   51 FORMAT(  33H0   END OF OUTPUT FOR CASE NUMBER, I7  )
   61 FORMAT(  31H0  TIME USED DURING CASE NUMBER,  I7,  5H  WAS,
     1         -2PF7.2,  23H  MINUTES.   A TOTAL OF, I6,
     2         31H  CASES HAVE NOW BEEN DONE.    , I4,  7H  CASES   /
     3         27H ARE LEFT TO BE DONE.  /  1H0  )
   65 FORMAT(  I6,  A6, I7,  A3, F4.0,A6, I6,  A5, I3,  A5, I6,  A5,
     1         I5,  A6, I5,  A5, I6,  A5, I3, I2,  A6, I5,  A6, I6,
     2         I6, I6, I2, 414  /  ( 100I, 5I4 )   )
   66 FORMAT(  I6,  A6, I7,  A3, F4.0,A6, I6,  A5, I3,  A5, I6,  A5,
     1         I5,  A6, I5,  A5, I6,  A5, I3, I2,  A6, I5,  A6, I6,
     2         I6,  A5, I3,  A6, I6,  A5  )
   81 FORMAT(  7AI, ONEMORE =, F7.3, F9.3, 9H = THCNTS              /
     1         35H0    THE CALCULATION FOR CASE NUMBER, I7, 10H  HAS BEEN
     2         24H TERMINATED.    TIME USED =, I3, 9H MINUTES., F9.3,
     3         9H SECONDS.    )
   82 FORMAT(  CPANTCTALS,I6,I6,A1,25X2HONEMORE =F8.6,F12.3,9H = THCNTS /
     1         1H0, 33X, 13H LINE NUMBER , I7X, I6,    I11,    2X, A10   )
   91 FORMAT(        34X, 13HMAXIMUM VALUE, 12X, F8.6, F12.3, 1X, 6F8.2 )
   94 FORMAT(  1H0  /  41, 34H ADDITIONAL OUTPUT FOR CASE NUMBER, I7    /
     1         91H0 NO.  CNT COSNT NSTP NUS  FLIGHT DISTANCE      BETA
     2         A1, 16HMACH NO  ACCEL  , A6, 6H ND E0, 6H A6, I1, 18 )   )
   95 FORMAT(  3I3,I6,I6, I1,,A6, F9.4, F8.3, F8.4, F12.3, 1X, 6F8.2    )
   96 FORMAT(  1H0, 40X, 31H VALUES  USED      BTAADS  =    , 6F8.5  )
   97 FORMAT(        40X, 31H TO  COMPUTE      CNDS    =    , 6F8.5  )
   98 FORMAT(        40X, 31H TEMPERATURES       E     =    , 6F8.5  )
   99 FORMAT(  1H0, 45X, 23H EQUILIBRIUM RATIOS ARE.     11X, 6F8.5  )
   96 FORMAT(  90X 41H MAXIMUM EQUILIBRIUM TEMPERATURES ARE     6F8.2  )
```

19


```
C     TOTALING OF NSTP AND NUSL
 811  NSTPTT = NSTPTT + NSTP
      NUSLTT = NUSLTT + NUSL

C     SPECIAL ENTRY POINT FOR OUTPUT CONTINUATION
      ENTRY OUTCNT

C     CONVERSION OF INTERMEDIATE VALUES
 8111 GAMDEG = GAMMA * DGRADR
      PSIDEG = PSI   * DGRADR
      DELDEG = DELTA * DGRADR
      THEDEG = THETA * DGRADR
      BETDEG = BETA  * DGRADR
      ACCEL  = SQRT( ACCEL2 )
      FMACHN = SQRT( FMACH2 )
      N      = N     + 1
      ISNTAL(N)   = ICNTRL
      ISKONT(N)   = KOUNT
      ISNSTP(N)   = NSTP
      ISNUSL(N)   = NUSL
      SB(N)  = BETDEG
      SA(N)  = ACCEL
      SM(N)  = FMACHN
      IF( FNDOUT ) 812, 813, 813
 812  ACCEL  = ENDOUT
      GO TO 819
 813  IF( FNGOUT ) 814, 815, 814
 814  ACCEL  = FNGOUT
      GO TO 819
 815  IF( FNNOUT ) 816, 817, 816
 816  ACCEL  = FNNOUT
      GO TO 819
 817  IF( SHKOUT ) 818, 821, 818
 818  ACCEL  = SHKOUT
 819  ENDOUT = 0.0
      ENGOUT = 0.0
      FNNOUT = 0.0
      SHKOUT = 0.0

 821  IF(   BOOL( NTP  )    ) 822, 831, 822
 822  TPNDEO       = TGNDEO ** 0.25
      STNDEO(N)    = TPNDEO
      IF(  NTP    ) 823, 825, 823
 823  DO 824   IT = 1, NTP
 824  STP(IT, N)  = TEMPS(1,IT)
 825  IF( ICNTRL -  899    ) 826, 827, 827
 826  TPEBOT =  TPNDEO
      GO TO 831
 827  IF( ICNTRL -  909    ) 828, 826, 826
 828  IT       = ITSAVE
      TPEBOT =  TEMPS(1,IT)

 831  SS(N)  =  AFOOKD( SD )
```



```
C     COMPUTATION OF INTERMEDIATE ENERGY VALUES
 841  IF( ENRCON ) 842, 842, 851
 842  CALL GRAVCO

C     OUTPUT OF INTERMEDIATE VALUES
 851  FXOUT  =  IFFORMAT  AFOAMSE( FX       )  )
      FYOUT  =  IFFORMAT  AFOAMSE( FY       )  )
      GAMOUT =  AFORMAT   AFOAMSE( GAMDEG )  )
      PSIOUT =  IFFORMAT  AFOAMSE( PSIDEG )  )
      DELOUT =  IFFORMAT  AFOAMSE( DELDEG )  )
      THEOUT =  IFFORMAT  AFOAMSE( THEDEG )  )
      TIMOUT =  IFFORMAT  AFOAMSE( TIME     )  )
      ALPOUT =  IFFORMAT  AFOAMSE( ALPHAD )  )
      PHIOUT =  IFFORMAT  AFOAMSE( PHIDEG )  )
      ACCOUT =  IFFORMAT  AFOAMSE( ACCEL    )  )
      TPEOUT =  IFFORMAT  AFOAMSE( TPEBOT )  )
      IF( 360.0  -  PHIDEG ) 852, 853, 853
 852  PHIDEG =  ATAN( SINPHI / COSPHI ) * DGRPRD
      PHIOLD =  PHIDEG
      PHIOUT =  IFFORMAT  AFOAMSE( PHIDEG )  )
 853  IF(   BOOL(GITVLS)    ) 854, 854, 854
 854  IF(        GITVLP     ) 855, 855, 856
 855  WRITE(6,885) FXOUT,FYOUT,GAMOUT,PSIOUT,DELOUT,THEOUT,TIMOUT,
     1             ALPOUT,PHIOUT,NOUT,ACCOUT,TPEOUT,ICNTRL,
     2             NSTP,NUSL,JFAIL,(IETCTL(I),I=1,JFAIL)
      CALL TPELST
      GO TO 899
 856  CALL RANGER
      DWNOUT =  IFFORMAT  AFOAMSE( DOWNR   )  )
      CRSOUT =  IFFORMAT  AFOAMSE( CROSSR )  )
      PSROUT =  IFFORMAT  AFOAMSE( PSIRG   )  )
      PSIRGA =  ABS( PSIRG * CONDEG )
      DTASPN =  DWSINC
      NOUT   =  N
      IF( I -  JFAIL  ) 857, 858, 899
 857  NOUT   =  ISIGN( NOUT, 1-NOUT )
 858  ICNTRL =  - ICNTRL
      ISNTAL(N)  =  TOTCTL
 859  WRITE(6,886) FXOUT,FYOUT,GAMOUT,PSIOUT,DELOUT,THEOUT,TIMOUT,
     1             ALPOUT,PHIOUT,NOUT,ACCOUT,TPEOUT,ICNTRL,
     2             DWNOUT,CRSOUT,PSROUT
      CALL TPELST

C     OUTPUT OF INTERMEDIATE VALUES SENSE SWITCH CONTROL
 890  IF(   SNSSOT    ) 8991,8993,8991
 8991 IF( SWITCH(890) ) 8992,8993,8992
 8992 AA = OUT
 8993 CONTINUE

C     NORMAL RETURN FROM SUBROUTINE OUTPUT
 891  RETURN
```



```
      ENTRY OUTFNL

C     SUBROUTINE OUTFNL CORE NUMBER OCTFNL

C     THIS PROGRAM IS DESIGNED TO OUTPUT CERTAIN FINAL VALUES OF THE
C     CALCULATION AS WELL THOSE VALUES WHICH WERE NOT IMMEDIATELY
C     OUTPUT ONTO THE OUTPUT TAPE

C     OUTPUT OF STORED VALUES
 911  TMFUSD =  CLOCK(TMF)
      TMF    =  AINT( (TMF+TMFUSD)*36.0 + 0.375 )  /  36.0
      TMEMAC =  AINT(  TMFUSD * 36.0  +  0.375  )
      MINTMF =  TMEMAC  /  3600.0
      SECTMF =  AMOD(  TMEMAC, 3600.0  )  /  60.0
      WRITE(6,911)  ENDURE,THCNTS,NOCASE,MINTMF,SECTMF
      SPACE  =  0.0
      IF( N  .GT.  10 ) SPACE = SEJECT
 912  IF(          SNATOT     ) 913, 951, 951
 913  IF(           NTP       ) 914, 933, 914
 914  WRITE(6,941)  SPACE,NOCASE,SOLANE,TPHEAD,ITPHEAD,(IT,IT=1,NTP)
      DO 915  K = 1, N
 915  WRITE(6,951)  K,ISNTAL(K),ISKONT(K),ISNSTP(K),ISNUSL(K),SS(K),
     1              SB(K),SM(K),SA(K),STNDEO(K),(STP(IT,K),IT=1,NTP)
      ACCMAX =  AMAXAF SA, IACMAX, 1, N, 1 )
      TPMOHE =  AMAXAF STNDEO, ITPMON, 1, N, 1 )
      DO 921   IT = 1, NTP
      TPMAX(IT)  =  AMAXAF STP, ITMAX, IT, 6NN, 6 )
      TPFOHE(IT) =  TPMOHE * PO(IT)
 921  ITPMAX(IT) =  ( ITMAX + 9 )  /  6
 922  WRITE(6,922)  NSTPTT,NUSLTT,SOLANE,ENDURE,THCNTS,IACMAX,ITPMON,
     1              (ITPMAX(IT),IT=1,NTP)
 923  WRITE(6,931)  ACCMAX,TPMOHE,(TPMAX(IT),IT=1,NTP)
 924  WRITE(6,961)  (OTHADS(IT),IT=1,NTP)
 927  WRITE(6,971)  (CMOS(IT),IT=1,NTP)
 928  WRITE(6,981)  (E(IT),IT=1,NTP)
 929  WRITE(6,991)  (PO(IT),IT=1,NTP)
 930  WRITE(6,991)  (TPFOHE(IT),IT=1,NTP)
      GO TO 951

 931  IF(   BOOL(SNATOT)     ) 933, 932, 933
 932  IF(   BOOL(THCNTS)     ) 941, 944, 941
 933  IF(   BOOL( NTP  )     ) 934, 932, 934
 934  WRITE(6,941)  SPACE,NOCASE,SOLANE,TPHEAD
      DO 935   K = 1, N
 935  WRITE(6,951)  K,ISNTAL(K),ISKONT(K),ISNSTP(K),ISNUSL(K),SS(K),
     1              SB(K),SM(K),SA(K),STNDEO(K)
      ACCMAX =  AMAXAF SA, IACMAX, 1, N, 1 )
      TPMOHE =  AMAXAF STNDEO, ITPMON, 1, N, 1 )
      WRITE(6,921)  NSTPTT,NUSLTT,SOLANE,ENDURE,THCNTS,IACMAX,ITPMON
      WRITE(6,931)  ACCMAX,TPMOHE
      GO TO 951
```



```
 941  WRITE(6,941)  SPACE,NOCASE,SOLANE
      DO 942  K = 1, N
 942  WRITE(6,951)  K,ISNTAL(K),ISKONT(K),ISNSTP(K),ISNUSL(K),SS(K),
     1              SB(K),SM(K),SA(K)
      ACCMAX =  AMAXAF SA, IACMAX, 1, N, 1 )
      WRITE(6,921)  NSTPTT,NUSLTT,SOLANE,ENDURE,THCNTS,IACMAX
      WRITE(6,931)  ACCMAX
      GO TO 951

 944  WRITE(6,941)  SPACE,NOCASE
      DO 945  K = 1, N
 945  WRITE(6,951)  K,ISNTAL(K),ISKONT(K),ISNSTP(K),ISNUSL(K),SS(K),
     1              SB(K)
      WRITE(6,921)  NSTPTT,NUSLTT

C     OUTPUT OF TOTAL AND PARTIAL ENERGIES THROUGH SUBROUTINE GRAVCS
 951  IF( ENRCON ) 952, 952, 953
 952  CALL GRAVCO
 953  WRITE(6,911)  NOCASE

C     KICK-OFF PROCEDURES AND OPERATOR INFORMATION ON-LINE PRINT OUT
 961  IF( SNSSOT ) 962, 964, 962
 962  IF( SWITCH(3)    ) 963, 964, 963
 963  CALL FINAL
      STOP
 964  CALL FNDLST
 965  IF( N .GT. NSTOP ) N = NSTOP

 971  CNTCSE =  CNTCSE            +   1
 972  CNTFNM =  IABS(CNTMAX)      -   CNTCSE
 973  IF( CNTMAX ) 974, 990, 975
 974  IF( SWITCH(4)    ) 990, 975, 990
 975  PRINT 81,     NOCASE,TMFUSD,CNTCSE,CNTFNM
      GO TO 990
 976  IF( SWITCH(4)    ) 975, 990, 975

 983  IF( SWITCH(4)    ) 981, 990, 981
 981  PRINT 81,     NOCASE,TMFUSD,CNTCSE

C     OUTPUT OF STORED VALUES SENSE SWITCH CONTROL
 990  IF(   SNSSOT    ) 9901,9903,9901
 9901 IF( SWITCH(990) ) 9902,9903,9902
 9902 AA = OUT
 9903 CONTINUE

C     NORMAL RETURN FROM SUBROUTINE OUTFNL
 991  RETURN

      END
```

```
$IBMAP SCCSPA  900,ND,DECK
*       SINE - COSINE SUBPROGRAM AND FUNCTION SUBROUTINES   VERSION 6.0
*       THESE ROUTINES ARE DESIGNED TO QUICKLY CALCULATE THE SINE AND
*       COSINE IN THE VICINITY OF  N * PIE/2 (FOR ALL N).    IN ADDITION,
*       THE SUBPROGRAM SUBROUTINES WILL FIRST CHECK TO SEE IF THE VALUE
*       OF THE ARGUMENT HAS CHANGED.
        LBL     SCCSPA,BEGIN,
        SPACE   4
        ENTRY   SNCSPH
SNCSPH  CLA     PHIDEG
        CAS     PHIOLD
        TRA     *+2
        TRA     3,4
        STO     PHIOLD
        SXD     SYSLOC,4
        LDQ     ZERO
        DFMP    RDMDEG
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        DST     SINPHI
        TRA     3,4
        SPACE   2
        ENTRY   SNCSAL
SNCSAL  CLA     ALPHAD
        CAS     ALPOLD
        TRA     *+2
        TRA     3,4
        STO     ALPOLD
        SXD     SYSLOC,4
        LDQ     ZERO
        DFMP    RDMDEG
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        DST     SINALP
        TRA     3,4
        SPACE   2
```



```
        ENTRY   SNCSGA
SNCSGA  DLD     FG
        CAS     FGOLD
        TRA     *+2
        TRA     3,4
        STO     FGOLD
        SXD     SYSLOC,4
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        DST     SING
        TRA     3,4
        SPACE   2
        ENTRY   SNCSPS
SNCSPS  DLD     FP
        CAS     FPOLD
        TRA     *+2
        TRA     3,4
        STO     FPOLD
        SXD     SYSLOC,4
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        DST     SINP
        TRA     3,4
        SPACE   2
        ENTRY   SNCSDE
SNCSDE  DLD     FD
        CAS     FDOLD
        TRA     *+2
        TRA     3,4
        STO     FDOLD
        SXD     SYSLOC,4
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        DST     SIND
        TRA     3,4
        SPACE   10
```



```
        ENTRY   SNCSDG
SNCSDG  CLA*    3,4
        TZE     ZROARG
        SXD     SYSLOC,4
        LDQ     ZERO
        DFMP    RDMDEG
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        TRA     4,4
        SPACE   2
        ENTRY   SNCSRA
SNCSRA  CLA*    3,4
        TZE     ZROARG
        SXD     SYSLOC,4
        LDQ     ZERO
        SXA     *+2,1
        TSX     SNCSRS,1
        AXT     **,1
        TRA     4,4
        SPACE   2
ZROARG  LDQ     ONE
        TRA     4,4
        SPACE   4
SNCSRS  DST     RADIAN
        SSP
CASP02  CAS     PIE304
        TRA     CASP
        TRA     CP02P
        TRA     FPIE02
CASP    CAS     PIE504
        TRA     CASPT2
        TRA     CPP
        TRA     FPIE
CASPT2  CAS     PIE904
        TRA     NODPT2
        TRA     CPT2P
        TRA     FPIET2
        SPACE   1
NODPT2  CAS     ANGLE2
        TRA     LARGE
        TRA     LARGE
        LDQ     RADIAN
        FMP     RIOV2P
        FAD     CHARDN
        DFMP    PIET2N
        DFAD    RADIAN
        TRA     SNCSRS
        SPACE   3
```



```
*       FURTHER TESTING ABOUT PIE/2
FPIE02  CAS     PIE104
        TRA     FCP02L
        TRA     C1P04
FCAP    CAS     ANGLEP
        TRA     C3P04
        AXT     0,0
C2P02   DLD     RADIAN
        FRN
        LDQ     ONE
        TRA     1,1
C1P04   DLD     RADIAN
        SXA     *+2,1
        TSX     CP04,1
        AXT     **,1
        LRS     COS
        TRA     1,1
FCP02L  CAS     PIE02L
        TRA     FCP02H
        TRA     SP02HL
CP02N   SSM
        DFAD    PIE02N
        CAS     ANGLEP
        TRA     C3P04
        AXT     0,0
        FRN
        STQ     EXTRA
CP02S   CLA     ONE
        LDQ     RADIAN
        LLS     9
        LRQ     EXTRA
        TRA     1,1
FCP02H  CAS     PIE02H
        TRA     CP02P
        AXT     0,0
SP02HL  STF     EXTRA
        TRA     CP02S
CP02P   SSM
        DFAD    PIE02P
        CAS     ANGLEP
        TRA     CP02S-2
        TRA     CP02S-2
C3P04   SXA     *+2,1
        TSX     CP04,1
        AXT     **,1
        STQ     COSRAD
        LDQ     RADIAN
        CLA     COS
        LLS     0
        LDQ     COSRAD
        TRA     1,1
        SPACE   5
```

21


```
*       FURTHER TESTING ABOUT PIE
PPIP    CAS     PIEL
        TRA     FCPH
        TRA     SPHL
CPH     DFSB    PIED
        CAS     ANGLEP
        TRA     CPS-1
        TRA     CPS-1
CSPA4   SXA     **2,1
        TSX     CPD4,1
        AXT     **,1
        CAS     RADIAN
        TRA     **3
        STO
        CHS
        XCA
        CLS     COS
        XCA
        TRA     1,1
FCPH    CAS     PIEH
        TRA     CPP
        AXT     0,0
SPHL    PAC
        LDQ     MONE
        TRA     1,1
CPP     DFSB    PIED
        CAS     ANGLEP
        TRA     CSPA4
        AXT     0,0
        FRN
CPS     CAS     RADIAN
        TRA     **3
        STO
        CHS
        LDQ     MONE
        TRA     1,1
        SPACE   20
```



```
*       FURTHER TESTING ABOUT 3PIE/2
PPI3T2  CAS     PIE7D6
        TRA     FCPT2L
        TRA     CPH3P
FCPH3L  CAS     PIEP3L
        TRA     FCPH3H
        TRA     SPH3HL
CPH3D   DFSA    PIEP3D
        CAS     ANGLEP
        TRA     CPH3S-2
        TRA     CPH3S-2
CTPA4   SXA     **2,1
        TSX     CPD4,1
        AXT     **,1
        STO     EXTRA
        CLS     RADIAN
        XCA
        CLA     CHS
        LLS     0
        LDQ     EXTRA
        TRA     1,1
FCPH3H  CAS     PIEP3H
        TRA     CPH3P
        AXT     0,0
SPH3H   STZ     COSRAD
        TRA     CPH3S
CPH3P   DFSA    PIEP3D
        CAS     ANGLEP
        TRA     CTPA4
        AXT     0,0
        FRN
        STO     COSRAD
CPH3S   LDQ     ONE
        CLS     RADIAN
        XCA
        LLS     0
        LDQ     COSRAD
        TRA     1,1
        SPACE   10
```



```
*       FURTHER TESTING ABOUT 2PIE
PCPT2L  CAS     PIET2L
        TRA     PCPT2H
        TRA     SPT2HL
CPT2D   DFSA    PIET2D
        CAS     ANGLEP
        TRA     CPT2S-1
        TRA     CPT2S-1
CSPA4   SXA     **2,1
        TSX     CPD4,1
        AXT     **,1
        CAS     RADIAN
        CHS
        AXT     0,0
        LDQ     CHS
        TRA     1,1
PCPT2H  CAS     PIET2H
        TRA     CPT2P
        AXT     0,0
SPT2HL  PAC
        LDQ     ONE
        TRA     1,1
CPT2P   DFSL    PIET2D
        CAS     ANGLEP
        TRA     CSPA4
        AXT     0,0
        FRN
CPT2S   CAS     RADIAN
        CHS
        AXT     0,0
        LDQ     ONE
        TRA     1,1
        SPACE   14
```



```
*       OUT OF RANGE ERROR MESSAGE AND ERROR TRACE PROVISIONS
LARGE   SXA     .OUTR.,4
        TSX     **3,,1
        PZE
        PZE     CODE
        LXD     SYSLOC,4
        CLA     ZERO
        TRA     1,1
CODE    DEC     12
        PZE     MESS1,,14
        PZE     MESS2,,7
MESS1   BCI     9,0     SIN(X) AND COS(X).   /X/ GREATER THAN OR EQUAL TO
        BCI     5, 2**25 RADIANS NOT ALLOWED.
MESS2   BCI     7,0     SET  SIN(X) = 0.0,  SET  COS(X) = 1.0
*       GENERAL STORAGE AND CONSTANTS
COSRAD  DEC     0.0
RADIAN  DEC     0.0,0.0
EXTRA   DEC     0.0
MZERO   DEC     -0.0
ZERO    DEC     0.0
ONE     DEC     1.0
MONE    DEC     -1.0
CHARAC  OCT     200000000000
BIGPIE  DEC     -0.15915494E5
PIEP3L  OCT     203622077325
PIEP3D  OCT     203622077327,150615641603
PIE7D6  OCT     200622077300
ANGLEP  OCT     561993346377
ANGLEP  OCT     161993346377
PIE1D6  OCT     200622077327
PIET2L  OCT     202622077300
PIE9D6  OCT     202766117217
PIE7D6  OCT     203517067177
PIE1D6  OCT     203706307200
ANGLEM  OCT     231400000000
        SPACE   1
```

22


```
CP04    DST     DELTAR
        DUFA    CHAPUA
        RQL     8
        LRS     0
        STO     DELTAI
        MPY     DELTAI
        ARS     1
        STO     PD2OV2
        XCA
        MPY     M1OV3
        CAS     DECON
        TRA     TERM2
        TRA     TERM2
CP04C4  STO     MD2OV3
        XCA
        MPY     PD2OV2
        ARS     1
        STO     MD4OV6
        XCA
        MPY     M1OV5
        CAS     D4CON
        TRA     CP04C6
        AXT     0,0
TERM4   SUM     MD2OV3
        ADM     PONE
        XCA
        CLS     MD4OV6
        TRA     CP04PN-2
CP04C6  STO     PD4OV5
        XCA
        MPY     MD2OV3
        STO     MD6OV6
        XCA
        MPY     P1OV7
        CAS     D6CON
        TRA     TERM6
        TRA     TERM6
CP04C8  STO     MD6OV7
        XCA
        MPY     PD2OV2
        XCA
        MPY     M1OV4P
        STO     PD8OV8
        XCA
        MPY     P1OV9P
TERM8   SUM     MD6OV7
        ADM     PD4OV5
        SUM     MD2OV3
        ADM     PONE
        XCA
        CLA     PD8OV8
        SUM     MD6OV6
        ADM     MD4OV4
        SUM     PD2OV2
        ADM     PONE
        SPACE   1
```



```
CP04PN  ADD     ROUND1
        ARS     8
        ORA     CHAR
        STQ     COS
        XCA
        LRS     8
        ADD     CHAR
        RQL     28
        DFMP    DELTAR
        FRN
        TRA     1,1
TERM6   ADM     PD4OV5
        SUM     MD2OV3
        ADM     PONE
        XCA
        CLA     MD6OV6
        TRA     CP04PN-3
TERM2   ADM     PONE
        XCA
        CLS     PD2OV2
        TRA     CP04PN-1
*       SPECIAL STORAGE AND CONSTANTS USED IN COMPUTING ABOUT PIE/4
COS     DEC     0
DELTAR  DEC     0.0,0.0
PD2OV2  DEC     0
DELTAI  DEC     0
MD2OV3  DEC     0
MD4OV4  DEC     0
MD6OV6  DEC     0
PD4OV5  DEC     0
MD6OV7  DEC     0
PD8OV8  DEC     0
        PZE
D2CON   DEC     -1010
ROUND1  OCT     201
D6CON   DEC     -1031
CHAR    OCT     200000000000
CHAPUA  OCT     233000000000,0
D4CON   DEC     1024
M1OV3   DEC     -0.333333333333333300
M1OV5   DEC     -0.200
P1OV7   DEC     0.142857142857143300
M1OV4P  DEC     -0.240420083940903400
P1OV9P  DEC     0.111740575373600000
PONE    OCT     377777777777
        SPACE   9
```



```
*       CONTROL SECTIONS FOR ARGUMENTS IN RADIANS
CPVAR   CONTRL  PVAR            NOTE  -  BINARY DECK WILL BE CHANGED TO
        USE     PVAR            PRIVIDE A DICTIONARY ORIGIN IN COMPLEX FORM
        DEC     0.0,0.0,0.0,0.0,0.0,0
FG      DEC     0.0,0.0
FP      DEC     0.0,0.0
FD      DEC     0.0,0.0
        DEC     0.0,0.0,0.0,0.0,0.0,0
        SPACE   1
CSCSPD  CONTRL  SCSPD           NOTE  -  BINARY DECK WILL BE CHANGED TO
        USE     SCSPD           PRIVIDE A DICTIONARY ORIGIN IN COMPLEX FORM
FGOLD   DEC     0.0
SING    DEC     0.0
COSG    DEC     1.0
FPOLD   DEC     0.0
SINP    DEC     0.0
COSP    DEC     1.0
FDOLD   DEC     0.0
SIND    DEC     0.0
COSD    DEC     1.0
        SPACE   2
*       CONTROL SECTIONS FOR ARGUMENTS IN DEGREES
CSCAP   CONTRL  SCAP            NOTE  -  BINARY DECK WILL BE CHANGED TO
        USE     SCAP            PRIVIDE A DICTIONARY ORIGIN IN COMPLEX FORM
ALPHAD  DEC     0.0
ALPOLD  DEC     0.0
SINALP  DEC     0.0
COSALP  DEC     1.0
PHIDEG  DEC     0.0
PHIOLD  DEC     0.0
SINPHI  DEC     0.0
COSPHI  DEC     1.0
        SPACE   2
```



```
*       CONTROL SECTIONS FOR COMMON CONSTANTS
CPIES   CONTRL  PIES            NOTE  -  BINARY DECK WILL BE CHANGED TO
        USE     PIES            PRIVIDE A DICTIONARY ORIGIN IN COMPLEX FORM
PIED    OCT     202622077325,147042055061    3.14159265358979323846
PIE02D  OCT     201622077325,146042055061    1.57079632679489661923
PIET2D  OCT     203622077325,150042055061    6.28318530717958647692
RDDDD   OCT     173435750650,140451235161    0.01745329251994329577
DDDRDD  OCT     206712273406,153461701757    57.29577951308232087680
RDDDR   OCT     173435750651                 0.01745329277
DDDRDR  OCT     206712273407                 57.2957790
PIE02L  OCT     201622077324
PIE02H  OCT     201622077326
PIEL    OCT     202622077324
PIEH    OCT     202622077326
PIET2L  OCT     203622077324
PIET2H  OCT     203622077326
        SPACE   34
        END
```

23


```
STOPTC DCVR    PULIST,PHASE,DO,DECK
      SUBROUTINE DENVAR

C     SUBROUTINE DENVAR CODE NUMBER DCNVAR                  VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO PRODUCE THE VARIABILITY FACTOR BY
C     WHICH THE DENSITY AS DERIVED FROM SUBROUTINE PTTDS (OR A SIMPLE
C     EXPONENTIAL WHEN STNDRD = - 0.01 IS TO BE MULTIPLIED.  THE
C     VARIABILITY FACTOR MAY BE ANY DEPENDENT COMBINATION OF ALTITUDE
C     OR RANDOM VARIATION ALONG THE FLIGHT PATH.

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  DDUMP(2),DEPTHS,NOCASE,DDUMO(9)
      COMMON /CVAR/    VDUMP(32),S,VDUMO(1)
      COMMON /CTVAR/   TDUMP(10),CTEMP,TDUMO(1)
      COMMON /CPVAR/   PDUMP(6),PY,PDUMO(19)
      COMMON /CAVAR/   ADUMP(30),DS,ADUMO(1)
      COMMON /CINTS/   M,N,NTP,IOPOYD,IONORE,ICOUNT,IDUMO(14)
      COMMON /CPIES/   PDUMP(4),PIET2,PDUMO(13)

C     COMMON STATEMENTS FOR DEPTHS VARIABLES
      COMMON /CO2PP/   RHO2R2,BETAY2

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CPTTDS/  TDENTY,PRESS,TMOL,TATM,DENSTY,SNDSPD,SDSPD2,VMOL,
     1                 YEAR,IPTDS1,IPTDS2,IPTDS3,IPTDS4

C     DOUBLE PRECISION STATEMENTS FOR LOCAL VARIABLES
      DOUBLE PRECISION BETA1P,BETA2P

C     DIMENSION STATEMENTS FOR LOCAL VARIABLES
      DIMENSION        B(4),FACOUT(10)
      DIMENSION        AMPVLS(32),AMPLDE(30),PYVLS(32),PYDEP(30)
      DIMENSION        SLPVLS(31),SLOPEA(30),CONVLS(31),CONSTA(30)

C     EQUIVALENCE STATEMENTS FOR LOCAL VARIABLES
      EQUIVALENCE      (AMPVLS(1),AMPLE2),(AMPVLS(2),AMPLDE)
      EQUIVALENCE      (PYVLS(1),PYDEP2),(PYVLS(2),PYDEP)
      EQUIVALENCE      (SLPVLS(2),SLOPEA),(CONVLS(2),CONSTA)

C     LOCAL DATA VALUES
      DATA   B,IREPNE,0200,0400,PYSPLT/0.,0.5,0.5,1.0,0.0200,0400,-1.06/
      DATA   STATIN,VARNCE,PRCMIN,PRCMAX,SINCR/.5,.025,-.05,.05,520000./
      DATA   AMPFIX,AMPSIN,WAVELN,PYMODE,IREP/0.1,0.1,50000.0,250000.,1/
      DATA   AMPLE2,AMPLEN,PYDEP2,PYDEPN/0.0,0.0,-1.0E9,1.0E10/
      DATA   AMPLDE(1),PYDEP(1),SLOPEA(1),CONSTA(1)/0.0,1.0E10,0.0,0.0/

C     FORMATS
   13 FORMAT( 30H0       ERROR =   DENVAR   I5), 5I15  )
```



```
C     COMPUTATION OF THE VARIABILITY FACTORS
  101 IF( PY  .NE.  PYOLD  ) GO TO 104
  102 IF(   BOOL(RANVAR)    ) 103, 199, 103
  103 DENSTY =  DENOLD
      GO TO 152
  104 IF(          STNDRD    ) 106, 105, 106
  105 IF( PY  .GT.  PYSPLT ) GO TO 1052
 1051 DENSTY =  RHO2R1  *  EXP( - PY * BETAY1 )
      GO TO 107
 1052 DENSTY =  RHO2R2  *  EXP( - PY * BETAY2 )
      GO TO 107
  106 CALL PTTDS
  107 PYOLD  =  PY
      DENOLD =  DENSTY
  108 IF(   BOOL(ALTVAR)    ) 109, 151, 109
  109 IF(          ALTVAR   ) 111, 121, 131

C     COMPUTATION OF THE ALTITUDE DEPENDENT VARIABILITY FACTOR
  111 FACTRA =  AMPSIN * SIN( AMOD((PY-PYMODE),WAVELN) / WAVELN * PIET2)
      IF( -1.0 .GT. ALTVAR ) FACTRA = FACTRA + AMPFIX
      IF(   BOOL(RANVAR)    ) 152, 141, 152

  120 IREP   =  IREP  -  1
  121 IF( PYDEP(IREP)    .GT. PY ) GO TO 120
  122 IF( PYDEP(IREP+1) .GT. PY ) GO TO 124
  123 IREP   =  IREP  +  1
      GO TO 122
  124 FACTRA =  SLOPEA(IREP) * PY  +  CONSTA(IREP)
      IF(   BOOL(RANVAR)    ) 152, 141, 152

  131 FACTRA =  AMPFIX
      IF(   BOOL(RANVAR)    ) 152, 141, 152

C     RE-CALCULATION OF THE DENSITY
  141 DENSTY =  ( 1.0 + FACTRA )  *  DENSTY
      GO TO 199
```



```
C     COMPUTATION OF THE RANDOM VARIABILITY FACTOR
  151 IF(   BOOL(RANVAR)    ) 152, 199, 152
  152 IF( M - 1  ) 153, 154, 156
  153 WRITE(6,13)  M,IOPOYD
      CALL EXIT
  154 IF( IOPOYD ) 155, 155, 155
  155 FS     =  STEMP
      GO TO 161
  156 IF( IOPOYD ) 155, 157, 155
  157 FS     =  S  +  B(M) * DS

  161 IF( FS   .LT.   SNEW ) GO TO 181
  162 FACOLD =  FACREP
      SLPEOL =  SLOPE
      SOLD   =  SREP
  163 FACREP =  FACNEW
      SREP   =  SNEW
  164 FACNEW =  AMIN1( AMAX1( RANDU(STARTG) * VARNCE, PRCMIN), PRCMAX )
      SLOPE  =  ( FACNEW - FACREP ) / SINCR
      SNEW   =  SNEW  +  SINCR
      GO TO 161

  172 IF( FS  .EQ.  FSOLD  ) GO TO 191
  173 FACTRR =  FACOLD  +  SLPEOL * ( FS - SOLD )
      GO TO 184

  181 IF( FS  .LT.  SREP   ) GO TO 172
  182 IF( FS  .EQ.  FSOLD  ) GO TO 191
  183 FACTRR =  FACREP  +  SLOPE  * ( FS - SREP )
  184 FSOLD  =  FS

C     RE-CALCULATION OF THE DENSITY
  191 DENSTY =  ( 1.0 + FACTRA + FACTRR )  *  DENSTY
  199 RETURN
```



```
C     INITIALIZATION ENTRY POINT FOR CHANGE OF CONSTANTS
      ENTRY CONSTS

C     FORMATS
   20 FORMAT(  F20.0, 2D20.0, F10.0, F2.0  )
   21 FORMAT(  F20.9, 3PD30.10, 1P3E15.7, 015  )
   22 FORMAT(  4F9.0, 5F10.0, I2  )
   23 FORMAT(  7F10.0  )

      IF( AMOD( DEPTHS, 0200   )  .EQ. 0.0 ) GO TO 201
      PYSPLT =  - 1.0D6
      IF( BETAY2 .LT. 0.5  ) GO TO 201
  200 READ (5,20)  RHO2R1,BETA1P,BETA2P,PYSPLT,TEST
      BETAY1 =  SNGL( 1.0D0 / BETA1P )
      BETAY2 =  SNGL( 1.0D0 / BETA2P )
      B0     =  RHO2R1  *  EXP( - 100000.0 * BETAY1 )
      B1     =  RHO2R1  *  EXP( - PYSPLT   * BETAY1 )
      B2     =  RHO2R2  *  EXP( - PYSPLT   * BETAY2 )
      B3     =  RHO2R2  *  EXP( - 390000.0 * BETAY2 )
      IF( TEST.EQ.1.0 ) WRITE(6,21)  RHO2R1,BETA1P,BETAY1,B0,B1,B1
      IF( TEST.EQ.1.0 ) WRITE(6,21)  RHO2R2,BETA2P,BETAY2,B2,B3,B2
  201 IF( AMOD( DEPTHS, 0400   ) ) 202, 204, 202
  202 READ (5,22)  STATIN,VARNCE,PRCMIN,PRCMAX,SINCR,
     1             AMPFIX,AMPSIN,WAVELN,PYMODE,IREPNE
      IREPNE =  MIN0( IREPNE, 30 )
  203 IF( IREPNE.GT.0 ) READ(5,23) ( AMPLDE(I), I=1,IREPNE )
      IF( IREPNE.GT.0 ) READ(5,23) ( PYDEP(I),  I=1,IREPNE )
  204 AMPLDE(IREPNE+1)  =  AMPLEN
      PYDEP(IREPNE+1)   =  PYDEPN
      IREPP1 =  IREPNE  +  1
      IREP   =  IREPNE / 2  +  1
      DO 205  I = 1, IREPP1
      SLOPEA(I-1)  =  ( AMPLDE(I)-AMPLDE(I-1) )/( PYDEP(I)-PYDEP(I-1) )
  205 CONSTA(I-1)  =  AMPLDE(I-1)  -  SLOPEA(I-1) * PYDEP(I-1)
  206 IF( STATOL.EQ.STATIN ) GO TO 209
      STATOL =  STATIN
  207 IF(   BOOL(RANVAR)    ) 208, 209, 208
  208 IF( STATOL.EQ.STR     ) STATOL = STATIN
      STARTG =  AMOD( STATOL, 1.0 )
      FACOLD =  AMIN1( AMAX1( RANDU(STARTG) * VARNCE, PRCMIN), PRCMAX )
      FACREP =  AMIN1( AMAX1( RANDU(STARTG) * VARNCE, PRCMIN), PRCMAX )
      FACNEW =  AMIN1( AMAX1( RANDU(STARTG) * VARNCE, PRCMIN), PRCMAX )
      SLPEOL =  ( FACREP - FACOLD ) / SINCR
      SLOPE  =  ( FACNEW - FACREP ) / SINCR
      SOLD   =  0.0
      SREP   =  SINCR
      SNEW   =  SINCR * 2.0
  209 RETURN
```

```
C     ENTRY POINT TO LIST RANDOM VALUES USED DURING PRENTRY
      ENTRY ENDLST

C     FORMATS
   26 FORMAT( 1H1, NOTE - THE FOLLOWING SET OF RANDOM VALUES WERE
     1 USED DURING CASE NUMBER, I7, 1H. / 1H0, 12X, 16HSTARTING VALUE =,
     2 F10.5, 12X 10HVARIANCE =F10.5,10X23HMINIMUM ALLOWED VALUE =F10.5/
     3  13X19HMAXIMUM ALLOWED VALUE =F10.5, 5X 23HCORRELATION INTERVAL =
     4 F11.1,5H  FEET / 1H0    POSITION5X26H R A N D O M    V A L U E
     5     N U M B E R  /  7X, 6H(FEET), 13X, 1H0, 9I10  /  1H  )
   27 FORMAT( F15.1, 5X, 10F10.5  )

  211 IF(   DBBL(RANVAR)   ) 212, 219, 212
  212 IF( STDTOL.LT.1.0    ) GO TO 219
  213 NOS    = SNEW / SINCR  +  1.0
      STARTS =  AMOD( STDTOL, 1.0 )
      SOUT   =  0.0
      SOUTT  =  0.0
  214 WRITE(6,26)  NOCASE,STARTS,VARNCE,PRCMIN,PRCMAX,SINCR,(I,I=1,9)
  215 DO 218  J = 1, NOS, 10
      DO 216  I = 1, 10
      K      =  I
      FACOUT(I) =AMIN1( AMAX1( RANDG(STARTS) * VARNCE, PRCMIN), PRCMAX )
      IF( SOUTT .EQ.  SNEW ) GO TO 217
  216 SOUTT  =  SOUTT  +  SINCR
  217 WRITE(6,27)  SOUT,(FACOUT(I),I=1,K)
  218 SOUT   =  SOUT   +  10.0 * SINCR
  219 RETURN

C     INITIALIZATION ENTRY POINT
      ENTRY GENVAR

  221 FACTRA =  0.0
      FACTRA =  0.0
      FYOLD  =  - 1.0
      FYOLD  =  - 1.0
      INEP   =  INEPMX / 2  +  1
  222 IF( TWENTY .LT. 10.0 ) GO TO 231
  223 IF( TWENTY .LT. 40.0 ) GO TO 301
  224 IF( TWENTY .LT. 70.0 ) GO TO 251
  229 IF( TWENTY .LT.100.0 ) GO TO 401

  231 STNDRD =  TWENTY
      YEAR   =  0.0
      RANVAR =  0.0
      CALL PTTOSI
      GO TO 904

  251 STNDRD =  - 0.0
      SDSPR2 =  1000000.0
      ALTVAR =  TWENTY  -  40.0
      GO TO 901
```



```
  301 STNDRD =  - 1.0
      YEAR   =  1959.0
      ALTVAR =  TWENTY  -  10.0
      CALL PTTOSI
      GO TO 901

  401 STNDRD =  1.0
      YEAR   =  1962.0
      ALTVAR =  TWENTY  -  70.0
      CALL PTTOSI

  901 IF(   DBBL(ALTVAR)   ) 911, 902, 911
  902 RANVAR =  0.0
  903 IF(   DBBL(RANVAR)   ) 200, 904, 200
  904 RETURN

  911 IF( ALTVAR .GT.  9.0 ) GO TO 913
  912 RANVAR =  0.0
      GO TO 921
  913 ALTVAR =  ALTVAR  -  10.0
      RANVAR =  1.0

  921 IF(   DBBL(ALTVAR)   ) 922, 903, 922
  922 IF( ALTVAR .EQ.  1.0 ) GO TO 931
      IF( ALTVAR .EQ.  2.0 ) GO TO 932
      IF( ALTVAR .EQ.  3.0 ) GO TO 933
      IF( ALTVAR .EQ.  4.0 ) GO TO 934
      ALTVAR =  0.0
      GO TO 903

  931 ALTVAR =  - 1.0
      GO TO 903
  932 ALTVAR =  - 0.0
      GO TO 903
  933 ALTVAR =  1.0
      GO TO 903
  934 ALTVAR =  - 2.0
      GO TO 903


      END
```

```
$IBFTC PCTD    FULIST,PMAP,DD,DECK
      SUBROUTINE PTTDS

C     SUBROUTINE PTTDS  CODE NUMBER PCTDS                      VERSION 6.0

C     THIS PROGRAM WILL CALCULATE THE PRESSURE, DENSITY, MOLECULAR
C     TEMPERATURE, ATMOSPHERIC MOLECULAR WEIGHT, ATMOSPHERIC
C     TEMPERATURE, THE SQUARE OF THE SOUND SPEED, AND THE SOUND SPEED.
C     IT DOES NOT COMPUTE THE LAST FOUR QUANTITIES UNLESS IT WAS
C     INSTRUCTED TO DO SO BY THE APPROPIATE WORDS IN THE INITIALIZATION
C     CALLING SEQUENCE STARTING AT PTTDS1.

C     COMMON, TYPE, AND EQUIVALENCE STATEMENTS

      COMMON /CONTRL/  NDUMF(2),GOPTNS,DDUMB(10)
      COMMON /CPVAR/   FDUMF(6), ALTUDE, FDUMB(19)
      COMMON /CPTTDS/  TDENTY,PRESS,TMLC,TATM,DENSTY,SNDSPD,SDSPD2,WMOL,
     1                 YEAR,IPTDS1,IPTDS2,IPTDS3,IPTDS4
      COMMON /CRANGE/  GITYLE,DCQUIM(4)

      DIMENSION  ABASE(22),  PLM(21),  TMO(21),  PBASE(21)
      DIMENSION  GAMMAT(11), GAMMAL(11), ISRGN(21)
      DIMENSION  VALUES(14), VOLUES(14), VZLUES(14)
      DIMENSION  VALDES(9), VALDEF(9), VALDEN(9), VZLDEF(9), VZLDEN(9)
      DIMENSION  ABASE9(12), PLM9(11), TMO9(11), PBASEP(11), PBASEM(11)
      DIMENSION  ABASE2(12), PLM2(11), TMO2(11), PB2SEP(21), PB2SEM(21)
      DIMENSION  CA1(13), C0(13), C1(13), C2(13), C3(13), C4(13), C5(13)

      EQUIVALENCE  (VALUES,A2     ),(VALUES(2),A3     ),(VALUES(3),A4     )
     1            ,(VALUES(4),A5  ),(VALUES(5),A6     ),(VALUES(6),ASPLT  )
     2            ,(VALUES(7),FALOW ),(VALUES(8),FAPTUP),(VALUES(9),ISRSPD)
     3            ,(VALUES(10),ISAUP),(VALUES(11),FAISR),(VALUES(12),FAMPY)
     4            ,(VALUES(13),TZERO),(VALUES(14),TMUP )
      EQUIVALENCE  (VALDES,FACTAL),(VALDES(2),FACTT9),(VALDES(3),FACTS0)
     1            ,(VALDES(4),FAC2SQ),(VALDES(5),PZERO ),(VALDES(6),PDWP  )
     1            ,(VALDES(7),RHOMO),(VALDES(8),SNDSCO),(VALDES(9),SND2CO)

      DATA    ABASE/1200,C,150000.0,160000.0,170000.0,190000.0,230000.0,
     1        300000.0,400000.0,500000.0,600000.0,700000.0/
      DATA    PLM/1100.0,0.02,0.015,0.01,0.007,0.005,0.004,0.0035,
     1        0.0026,0.0017,0.0011/
      DATA    TMO/1100.0,360.65,960.65,1110.65,1210.35,1350.65,1550.65,
     1        1830.65,2160.65,2420.65,2590.65/
      DATA    FAUPP/700000.0/
```



```
C     COMPUTATION OF PRESSURE, DENSITY, AND MOLECULAR TEMPERATURE
 101  IF( ALTUDE -  ALTOLD ) 102, 174, 102
 102  ALTOLD =  ALTUDE
      FA     =  ALTUDE  *  FACTAL
 103  IF( FA -  FAUPP ) 104, 110, 194
 104  IF( FA -  ASPLT ) 105, 115, 106
 105  IF( FA          ) 106, 116, 197
 106  IF( FALOW -  FA ) 107, 117, 191
 107  FA     =  ((((A6*FA + A5)*FA + A4)*FA + A3)*FA + A2)*FA* FA  +  FA
 108  IF(    IALTST    ) 109, 112, 109
 109  ISR    =  MIN1(  FA * FAMPY, FAISR  )

 111  ISR    =  ISR  +  1
 112  IF( FA  -  ABASE(ISR+1)    ) 113, 121, 111
 113  IF( FA  -  ABASE(ISR)      ) 114, 122, 123
 114  ISR    =  ISR  -  1
      GO TO 113

 115  PRESS  =  PBASE(9)
      TMOL   =  TMO(9)
      ISR    =  8
      GO TO 128
 116  PRESS  =  PZERO
      TMOL   =  TZERO
      ISR    =  1
      GO TO 128
 117  PRESS  =  PBASE(1)
      TMOL   =  TMO(1)
      FA     =  ABASE(1)
      ISR    =  [illegible]
      GO TO 128
 110  PRESS  =  [illegible]
      TMOL   =  [illegible]
      FA     =  FAPTUP
      ISR    =  ISRUP
      GO TO 128

 121  ISR    =  ISR  +  1
 122  PRESS  =  PBASE(ISR)
      TMOL   =  TMO(ISR)
      GO TO 128
 123  HDIFF  =  FA  -  ABASE(ISR)
      TMOL   =  HDIFF  *  [illegible]          TMO(ISR)
 124  IF(-ISRGN(ISR)) 125, 126, 127
 125  PRESS  =  [illegible]
     1        [illegible]
     2        [illegible]
     3        [illegible]
      GO TO 128
 126  PRESS  =  EXP( HDIFF  [illegible]  )  [illegible]
      GO TO 128
 127  PRESS  =  (  TMOL / TMO(ISR)     ** GAMMAL(ISR)      [illegible]
 128  DENSTY =  PRESS  /  TMOL  *  [illegible]
 129  TMLC   =  FACTT9  *  TMOL
```



```
C     COMPUTATION OF A[illegible]    [illegible]RIC MOLECULAR WEIGHT, ATMOSPHERIC
C     TEMPERATURE, TH[illegible]  [illegible]E OF THE SOUND SPEED, AND THE SOUN[illegible]   [illegible]
 131  IF(   ITMORG     ) 132, 171, 132
 132  IF(   ITATMO     ) 141, 155, 151

 141  IF( 8  -   ISR  ) 142, 147, 147
 142  IF( FA-100000.0 ) 143, 145, 144
 143  WMOL   =  27.000  +  ATAN(  (220000.0-FA)/ 25000.0 ) * 5.01403570
 144  TATM   =  ( WMOL / 28.965  ) * TMLC
      GO TO 161
 145  WMOL   =  27.106
      TATM   =     0.935786700       * TMLC
      GO TO 161
 146  WMOL   =  27.106  +  ATAN(  (100000.0-FA)/1600000.0 ) * 7.03560710
      GO TO 144
 147  WMOL   =  28.966
      TATM   =  TMLC
      GO TO 161
 151  IF( 8  -   ISR   ) 152, 157, 157
 152  IF(  ISR  -  14  ) 157, 153, 156
 153  WMOL   =  ((((( 1.61068000000E-20  * FA  -  1.11608041300E-22)* FA  +
     1                 4.59201414010E-17 )* FA  -  6.04492130500E-12)* FA  +
     2        5.65219103500E-07 )* FA  -  2.75300384200E-02)* FA  +  376.997101
 154  WMOL   =     0.999944743       * WMOL
      TATM   =  ( WMOL / 28.9644 ) * TMLC
      GO TO 161
 155  WMOL   =  0.0
      TATM   =  0.0
      GO TO 161
 156  WMOL   =  ((((( 1.10095400000E-33  * FA  -  3.62016200000E-27)* FA  +
     1                 5.01596000000E-21 )* FA  -  2.04012900000E-15)* FA  +
     2        5.64221410000E-10 )* FA  -  1.07004000000E-04)* FA  +  35.6200050
      GO TO 154
 157  WMOL   =  28.9644
      TATM   =  TMLC

 161  IF(   ISDSPD     ) 162, 172, 175
 162  IF( ISR -ISRSPD ) 163, 163, 164
 163  SDSPD2 =        TMOL  * FAC2SQ
      GO TO 173
 164  SDSPD2 =  SND2CO
      GO TO 173
 171  WMOL   =  0.0
      TATM   =  0.0
 172  SDSPD2 =  0.0
 173  SNDSPD =  0.0
 174  RETURN
 175  IF( ISR -ISRSPD ) 176, 176, 177
 176  SDSPD2 =        TMOL  * FAC2SQ
      SNDSPD =  SQRT(TMOL) * FACTS0
      GO TO 174
 177  SDSPD2 =  SND2CO
      SNDSPD =  SNDSCO
      GO TO 174
```



```
C     OUT-OF-SPECIFIED RANGE SECTIONS

  93  FORMAT(  93H0    THIS IS AN ERROR MESSAGE FROM SUBROUTINE PTTDS -/
     1  103H      THE ALTITUDE IS OUTSIDE THE VALID RANGE AND ATMOSPHERIC
     2 PROPERTIES CANNOT, THEREFORE, BE COMPUTED./ 50H     THE VALUE OF
     3THE ALTITUDE TO BE USED WAS EQUAL TO, F14.3, A8, 1H. )

 101  ISR    =  1
      GO TO 106
 104  ISR    =  ISRUP
 105  IF(       A5      ) 106, 123, 106
 106  FA     =  ((((A6*FA + A5)*FA + A4)*FA + A3)*FA + A2)*FA* FA  +  FA
      GO TO 123

 191  IF( BOOL(AUTRG) ) 192, 101, 192
 192  IF(      AUTRG   ) 193, 117, 197
 193  WRITE(6,93)  ALTUDE, IDMEN
      GO TO 117
 194  IF( BOOL(AUTRG) ) 195, 104, 195
 195  IF(      AUTRG   ) 196, 110, 197
 196  WRITE(6,93)  ALTUDE, IDMEN
      GO TO 110
 197  WRITE(6,93)  ALTUDE, IDMEN
      CALL EXIT
      STOP
```

```
C     INITIALIZATION ENTRY POINT
      ENTRY PTTABI
C     INITIALIZATION OF COMPUTATION INSTRUCTIONS
      EQUIVALENCE (IOMEN,IPTOS(1)),(IALTS,IPTOS(2)),(IUNCH,IPTOS(3))
      EQUIVALENCE (IERROR,IPTOS(4))
      EQUIVALENCE (V1VLUES,A190 ),(V1VLUES(2),A390 ),(V1VLUES(3),A490 )
     1     ,(V1VLUES(4),A590 ),(V1VLUES(5),A690 ),(V1VLUES(6),ASPLT9)
     2     ,(V1VLUES(7),PVLOW ),(V1VLUES(8),P90TUP),(V1VLUES(9),I90SPD)
     3     ,(V1VLUES(10),I90UP),(V1VLUES(11),P9(SD),(V1VLUES(12),P9NPV)
     4     ,(V1VLUES(13),T2E09),(V1VLUES(14),THOUP)
      EQUIVALENCE (V1LDEF,P9CTPT),(V1LDEF(2),P9CTNE),(V1LDEF(3),P9CTSP)
     1     ,(V1LDEF(4),P9C2SP),(V1LDEF(5),PIEROP),(V1LDEF(6),P9NUPP)
     2     ,(V1LDEF(7),9NNUP),(V1LDEF(8),S9PSCD),(V1LDEF(9),S9P2CD)
      EQUIVALENCE (V1LDEN,P9CTAN),(V1LDEN(2),P9CTAL),(V1LDEN(3),P9CTSN)
     1     ,(V1LDEN(4),P9C2SN),(V1LDEN(5),PIEROE),(V1LDEN(6),P9NUPN)
     2     ,(V1LDEN(7),9NNUPN),(V1LDEN(8),S9NSCD),(V1LDEN(9),S9N2CD)
      EQUIVALENCE (V2LUES,A262 ),(V2LUES(2),A362 ),(V2LUES(3),A462
     1     ,(V2LUES(4),A562 ),(V2LUES(5),A662 ),(V2LUES(6),ASPLT2
     2     ,(V2LUES(7),P2LOW ),(V2LUES(8),P2OT [illegible] V2LUES(9),I2ESPD)
     3     ,(V2LUES(10),I2NUP),(V2LUES(11),P21    ,(V2LUES(12),P2NPV)
     4     ,(V2LUES(13),T2E02),(V2LUES(14),INOUP)
      EQUIVALENCE (V2LDEF,P2CTPT),(V2LDEF(2),P2CTNE),(V2LDEF(3),P2CTSP)
     1     ,(V2LDEF(4),P2C2SP),(V2LDEF(7),PIER2P),(V2LDEF(6),P2NUPP)
     2     ,(V2LDEF(7),9N2NUP),(V2LDEF(8),S2PSCD),(V2LDEF(9),S2P2CD)
      EQUIVALENCE (V2LDEN,P2CTAN),(V2LDEN(2),P2CTAL),(V2LDEN(3),P2CTSN)
     1     ,(V2LDEN(4),P2C2SN),(V2LDEN(5),PIER2N),(V2LDEN(6),P2NUPN)
     2     ,(V2LDEN(7),9N2NUN),(V2LDEN(8),S2NSCD),(V2LDEN(9),S2N2CD)

      DATAYEAR59,YEAR62,ALYEAR,YEAROL/1959.0,1962.0,9H YEAR,1959.0/
      DATASECOND,THIRD,FOURTH,FIFTH/6HSECOND,5HTHIRD,6HFOURTH,5HFIFTH/
      DATAIOMENP,IOMENN,IALTSP,IALTSC/[illegible]/
      DATAIUNCHN,IUNCHT,IUN2HS,IUN2HB/[illegible]/
      DATAIUNCHS,IUNCHB/[illegible]/
      DATAIERRPN,IERRPE,IERRPR,IERROR/[illegible]/

   71 FORMAT(52H0 THIS IS AN ERROR MESSAGE FROM SUBROUTINE PTTOSI -/40H
     1  AN INVALID COMMAND HAS BEEN GIVEN.  INSTEAD OF5X(3H  -A6,6H-  .))
   72 FORMAT( TH  OF  -A6, 9H-  , THE A6, 19H COMMAND WAS  -A6, 6H-  . )
   73 FORMAT(  9H  NOTE -  UNTIL OTHERWISE INSTRUCTED, ALL SUBSEQUENT
     1COMPUTATION WILL BE CARRIED OUT AS IF THE A6/17H   COMMAND WAS  -,
     2 A6, 6H-  .)
   74 FORMAT(1H+F65.0,1PE15.7,016  )
   92 FORMAT(  13A6, A2  )

      IEXIT  =  1
  201 IF( IALTS  -  IALTSP ) 202, 205, 202
  202 IF( IALTS  -  IALTSC ) 203, 204, 203
  203 WRITE(6,71)  IALTSP
      WRITE(6,72)  IALTSC, THIRD, IALTS
      WRITE(6,73)  IALTSC
      IEXIT  =  0
  204 IALTST =  1
      GO TO 211
  205 IALTST =  0
```



```
  211 IF( IUNCH  -  IUNCHN ) 212, 221, 212
  212 IF( IUNCH  -  IUNCHT ) 213, 222, 213
  213 IF( IUNCH  -  IUN2HS ) 214, 223, 214
  214 IF( IUNCH  -  IUN2HB ) 215, 224, 215
  215 IF( IUNCH  -  IUNCHS ) 216, 225, 216
  216 IF( IUNCH  -  IUNCHB ) 217, 226, 217
  217 WRITE(6,71)  IUNCHN, IUNCHT, IUN2HS, IUN2HB, IUNCHS
      WRITE(6,72)  IUNCHB, FOURTH, IUNCH
      WRITE(6,73)  IUNCHN
      IEXIT  =  0

  221 ITPRAN =  0
      GO TO 231
  222 ITATMP =  1
      ISCSPD =  0
      GO TO 228
  223 ITATMP =  0
      ISOSPD =  -1
      GO TO 228
  224 ITATMP =  1
      ISOSPD =  -1
      GO TO 228
  225 ITATMP =  0
      GO TO 227
  226 ITATMP =  1
  227 ISOSPD =  1
  228 ITPRAN =  1

  231 IF( IERROR  -  IERRPN ) 232, 239, 232
  232 IF( IERROR  -  IERRPE ) 233, 238, 233
  233 IF( IERROR  -  IERRPR ) 234, 237, 234
  234 IF( IERROR  -  IERRPA ) 235, 236, 235
  235 WRITE(6,71)  IERRPN, IERRPE, IERRPR
      WRITE(6,72)  IERRPA, FIFTH, IERROR
      WRITE(6,73)  IERRPN
  236 OUTDG  =  -1.0
      GO TO 241
  237 OUTDG  =   1.0
      IF(       IEXIT          ) 294, 294, 241
  238 OUTDG  =  -0.0
      GO TO 241
  239 OUTDG  =   0.0

  241 IF( YEAR   -  YEAROL ) 243, 242, 243
  242 IF( IOMEN  -  IOMNOL ) 243, 174, 243
  243 YEAROL =  YEAR
      IOMNOL =  IOMEN

  251 IF( YEAR   -  1959.0 ) 252, 301, 252
  252 IF( YEAR   -  1962.0 ) 253, 401, 253
  253 WRITE(6,71)  YEAR59
      WRITE(6,72)  YEAR62, ALYEAR
      WRITE(6,74)  YEAR, YEAR, YEAR
  254 GO TO 301
```



```
C     INITIALIZATION OF VALUES BASED ON 1959 US STANDARD ATMOSPHERE

  301 ABASE(12)  =  ABASE9(12)
      ITATMR     =  - ITATMR

      DO 327  I = 1, 14
  302 VALUES(I)  =  VALUES(I)
  303 IF( I      -      11 ) 304, 304, 327
  304 ABASE(I)   =  ABASE9(I)
  305 PLR(I)     =  PLR9(I)
  306 TMR(I)     =  TMR9(I)

  311 IF(     PLR(I)         ) 312, 313, 312
  312 GMROPL(I)  =  GMROP9 / PLR(I)
      ISRGRN(I)  =  - 1
      GO TO 314
  313 GMROTI(I)  =  GMROR9 / TMR(I)
      ISRGRN(I)  =     0
  314 CONTINUE

  321 IF( IOMEN  -  IOMENP ) 322, 323, 322
  322 IF( IOMEN  -  IOMENN ) 371, 325, 371
  323 PBASE(I)   =  PBASEP(I)
      IF( I      -       9 ) 324, 324, 327
  324 VALDES(I)  =  VALDEF(I)
      GO TO 327
  325 PBASE(I)   =  PBASEN(I)
      IF( I      -       9 ) 326, 326, 327
  326 VALDES(I)  =  VALDEN(I)
  327 CONTINUE

  331 FA     =  0.0
      ALTOLG =  0.0
      GO TO 116

  371 WRITE(6,71)  IOMENP
      WRITE(6,72)  IOMENN, SECOND, IOMEN

  381 IF( BOOL(GOPTN))     ) 382, 391, 382
  382 CALL GOPTNC

  391 IF( AMOD( GITVLE, 2.0 )   ) 399, 392, 399
  392 READ (5,92)  VALUES
  393 WRITE(6,92)  VALUES

  399 CALL EXIT
      STOP
```



```
C     DATA VALUES BASED ON 1959 US STANDARD ATMOSPHERE

      DATAABASE9/-5000.0,11000.0,25000.0,47000.0,53000.0,79000.0,
     1 90000.0,105000.0,160000.0,170000.0,200000.0,700000.0/
      DATAPLR9/-0.0065,0.0,0.003,0.0,-0.0045,0.0,0.004,0.02,0.01,0.005,
     1 0.0035/
      DATATMR9/320.66,216.66,216.66,282.66,282.66,165.66,165.66,225.66,
     1 1325.66,1425.66,1575.66/
      DATAA259,A359,A459,A559,A659/[illegible]/
C     DATAA259,A359,A459,A559,A659/-1.97312620E-07,2.4656995E-14,
C    1 -5.0667095E-21,6.04213348-28,-4.50136496E-35/
      DATAASPLT9,P9LOW,P90TUP,I90SPD,I90UP,P9(SD,P9NPV,T2E09,THOUP/
     1 700000.0,-4996.07035,430514.33,5,11,9.0,8.6E-5,200.16,3087.53714/
      DATAGOMRR9/-0.0341647942/
C     GOMRR9 =  - 9.80665  *  28.966  /  8314.39

      DATAPBASEP/[illegible]/
     1 [illegible]
     2 [illegible]/
      DATAP9CTPT,P9CTNE,P9CTSP,P9C2SP,PIEROP,P9NUPP,9NNUPP,S9PSCD,S9P2CD
     1 /[illegible]
     2 [illegible]/
C     P9CTPT =   0.3048
C     P9CTSP =  SQRT(  1.4 * 8314.39 / 28.966  )   /  0.3048
C     P9C2SP =       (  1.4 * 8314.39 / 28.966  )   /  0.3048 ** 2
C     9NNUPP =  28.966  /  8314.39  *  0.3048 ** 2
C     S9PSCD =  SQRT(  1.4 * 8314.39 / 28.966 * 105.66  )   /  0.3048
C     S9P2CD =  (  1.4 * 8314.39 / 28.966 * 105.66  )   /  0.3048 ** 2

      DATAPBASEN/[illegible]/
     1 [illegible]
     2 [illegible]/
      DATAP9CTAN,P9CTAL,P9CTSN,P9C2SN,PIEROE,P9NUPN,9NNUPN,S9NSCD,S9N2CD
     1 /1.0,1.0,20.0463335,401.899406,101325.0,0.[illegible]
     2 [illegible]/
C     P9CTSN =  SQRT(  1.4 * 8314.39 / 28.966  )
C     P9C2SN =         1.4 * 8314.39 / 28.966
C     9NNUPN =  28.966  /  8314.39
C     S9NSCD =  SQRT(  1.4 * 8314.39 / 28.966 * 105.66  )
C     S9N2CD =         1.4 * 8314.39 / 28.966 * 105.66
```

```
C     INITIALIZATION OF VALUES BASED ON 1962 US STANDARD ATMOSPHERE
 401  ABASE(12) = ABASE2(12)
      DO 427  I = 1, 31
      IF( I        -        14 ) 402, 402, 415
 402  VALUES(I)  =  V2LUES(I)
 403  IF( I        -        11 ) 404, 404, 415
 404  ABASE(I)   =  ABASE2(I)
 405  PLM(I)     =  PLM2(I)
 406  TMB(I)     =  TMB2(I)

 411  IF( I        -        9 ) 412, 415, 415
 412  IF(      PLM(I)         ) 413, 414, 413
 413  GOMDEL(I)  =  GOMOR2 / PLM(I)
      ISPARN(I)  =  - 1
      GO TO 416
 414  GOMDEL(I)  =  GOMOR2 / TMB(I)
      ISPARN(I)  =     0
      GO TO 416
 415  ISPARN(I)  =  1
C     NOTE -  THE FOLLOWING COMMENT CARDS SHOW THE FORMULAS BY WHICH THE
C     C VALUES WERE COMPUTED AND THE ACTUAL VALUES USED.  THE VALUES OF
C     CM1 TO C5 HAVE HAVE ALREADY BEEN SET VIA THEIR DATA STATEMENTS.
C     DATA     AA,AB,AC,AD,AE,AF/1.0,-1.5723000E-07,2.4650619E-14,
C    1         -3.0547346E-21,5.0173260E-28,-6.6505079E-35/
C     A          =    ABAS (I)   -   TMB(I)  /  PLM(I)
C     C5(I-0)    =    GOMOR2 / PLM(I)**6 *        6.0*AF               / 5.0
C     C4(I-0)    =    GOMOR2 / PLM(I)**5 * (   30.0*AF *A+ 5.0*AE) / 4.0
C     C3(I-0)    =    GOMOR2 / PLM(I)**4 * ((  60.0*AF *A+20.0*AE)*A+
C    1                                                      4.0*AD) / 3.0
C     C2(I-0)    =    GOMOR2 / PLM(I)**3 * ((( 60.0*AF *A+30.0*AE)*A+
C    1                                           12.0*AD)*A+ 3.0*AC) / 2.0
C     C1(I-0)    =    GOMOR2 / PLM(I)**2 * ((((30.0*AF  *A+20.0*AE)*A+
C    1                                 12.0*AD)*A+ 6.0*AC)*A+ 2.0*AB)
C     C0(I-0)    =  -  (((( C5(I-0)     * TMB(I) + C4(I-0)  ) * TMB(I) +
C    1 C3(I-0)  ) * TMB(I) + C2(I-0)  ) * TMB(I) + C1(I-0)  ) * TMB(I)
C     CM1(I-0)   =    GOMOR2 / PLM(I)     * (((((6.0*AF *A+ 5.0*AE)*A+
C    1                               4.0*AD)*A+ 3.0*AC)*A+ 2.0*AB)*A+AA)
 416  CONTINUE

 421   IF IOMEN   -   IOMENF ) 422, 423, 422
 422   IF IOMEN   -   IOMENH ) 371, 425, 371
 423  ABASE(I)   =  P02SEP(I)
      IF I       -          9 ) 424, 424, 427
 424  VALUES(I)  =  V2LDEF(I)
      GO TO 427
 425  ABASE(I)   =  P02SEN(I)
      IF( I      -          9 ) 426, 426, 427
 426  VALUES(I)  =  V2LDEN(I)
 427  CONTINUE

 431  GO TO 331
```



```
C     DATA VALUES BASED ON 1962 US STANDARD ATMOSPHERE

      DATAABASE2/-5000.0,11000.0,20000.0,32000.0,47000.0,52000.0,
     1 61000.0,79000.0,90000.0,100000.0,110000.0,120000.0/
      DATAPLM2/-0.0065,0.0,0.001,0.0028,0.0,-0.002,-0.004,0.0,0.003,
     1 0.005,0.01/
      DATATMB2/320.65,216.65,216.65,228.65,270.65,270.65,252.65,180.65,
     1 180.65,210.65,260.65/
      DATAA2C2,A3C2,A4C2,A5C2,A6C2/0992921672205,0123073716704,
     1 0479426261116,0.0,0.0/
C     DATAA2C2,A3C2,A4C2,A5C2,A6C2/-1.5733017E-07,2.3641629E-14,
C    1 -1.6018780E-21,0.0,0.0/
      DATAASPLT2,F2LOW,F2PTUP,T2PSPD,T2PUP,F2ISA,F2MPY,T2EP2,TMEUP/
     1 90000.0,-4996.0499,709000.0,7.21,10.0,0.0001,288.15,2700.65/
      DATAGOMOR2/-0.0341631947/
C     GOMOR2 =  - 9.80665  *  28.9644  /  8314.32

      DATAP02SEP/3711.96065,472.600406,114.349420,10.1209473,2.31632635,
     1 1.23226078,0.300323290,0.0216790606,0.00343319694,0.000620131700,
     2 0.000153602939,0.0000526660181,0.0000185710075,0.00000771904052,
     3 0.00000503258747,0.00000351971329,0.00000149374571,
     4 0.00000039349466,0.00000000441779900,0.00000000220043190,
     5 0.0000000072610404/
      DATAF2CTFT,F2CTRR,F2CT5P,F2C25P,PI0R2P,P02UPP,AMPMAP,S2P5CO,S2P2CO
     1 /0.1774700727767,1.0,65.7703503,6325.73007,2116.21662,
     2 0.000030007240911967,0.000323644124,003,993634,701444,705/
C     F2CTFT =   0.3048
C     F2CT5P =  SQRT(  1.4 * 8314.32 / 28.9644  )   /  0.3048
C     F2C25P =       (  1.4 * 8314.32 / 28.9644  )   /  0.3048 ** 2
C     AMPMAP =  28.9644  /  8314.32  *  0.3048 ** 2
C     S2P5CO =  SQRT(  1.4 * 8314.32 / 28.9644 * 100.65  )   /  0.3048
C     S2P2CO =  (  1.4 * 8314.32 / 28.9644 * 100.65  )   /  0.3048 ** 2

      DATAP02SEN/177606.075,22632.0640,5476.00060,060.010400,110.906300,
     1 59.0700400,18.2100737,1.03771176,0.104300443,0.0300791003,
     2 0.00739494045,0.00252170009,0.000906100070,0.000369496026,
     3 0.000279369790,0.000160525741,0.0000699697213,0.0000100300931,
     4 0.00000403046396,0.00000100570717,0.000000340030226/
      DATAF2CTRN,F2CTRL,F2CT5N,F2C25N,PI0R2N,P02UPN,AMPMAN,S2N5CO,S2N2CO
     1 /1.0,1.0,20.0468020,401.874301,101325.0,0.0000000119179606,
     2 0.00340367636,269.441260,72990,9924/
C     F2CT5N =  SQRT(  1.4 * 8314.32 / 28.9644  )
C     F2C25N =         1.4 * 8314.32 / 28.9644
C     AMPMAN =  28.9644  /  8314.32
C     S2N5CO =  SQRT(  1.4 * 8314.32 / 28.9644 * 100.65  )
C     S2N2CO =         1.4 * 8314.32 / 28.9644 * 100.65
```



```
      DATA CM1/0004991010163,
     1        [illegible],
     2        [illegible],
     3        [illegible]/

      DATA  C0/0597664094570,
     1        [illegible],
     2        [illegible],
     3        [illegible]/

      DATA  C1/0167666611967,
     1        [illegible],
     2        [illegible],
     3        [illegible]/

      DATA  C2/ 590612671714,
     1        [illegible],
     2        [illegible],
     3        [illegible]/

      DATA  C3/0112462990343,
     1        [illegible],
     2        [illegible],
     3        [illegible]/

      DATA  C4/0511719642990,
     1        [illegible],
     2        [illegible],
     3        [illegible]/

      DATA  C5/0079432730601,
     1        [illegible],
     2        [illegible],
     3        [illegible]/
```

END

```
$IOFTC RCRG    FULIST,PMAP,DD,DECK
      SUBROUTINE RANGE

C     SUBROUTINE RANGE  CODE NUMBER RANGE                        VERSION 6.0

C     THIS PROGRAM WILL CALCULATE THE EQUIVALENT DOWN RANGE, CROSS
C     RANGE, AND AZIMUTH CHANGE FOR MOVEMENT OVER THE EARTH BETWEEN ANY
C     PAIR OF INITIAL AND FINAL POINTS.

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CVAR/    VDUMP(10),PSII,VDUMP,DELTAI,VDUMD,THETAI,VDUMT,
     1                 VDUMB(30)
      COMMON /CFVAR/   FDUMP(12),PSI,FDUMP,DELTA,FDUMD,THETA,FDUMT,
     1                 FDUMB(8)
      COMMON /CSCOPO/  SCDUMP(7),SINDEL,COSDEL
      COMMON /CINTS/   IDUMP(7),ILAST,IDUMB(12)
      COMMON /CPIES/,  PIE,PDUM,PIE2,PDUMB,PIET2,PDUMT2,PDUMB(4),
     1                 DEGRAD,RADDEG,PIEO2L,PIEO2H,PIEH,PIEL,PDUMB(2)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CRANGE/  DITVLT,DDUMM,CDUESE,PSIAG,PSIMGA

C     MISCELLANEOUS TYPE AND DIMENSION STATEMENTS FOR LOCAL VARIABLES
      DIMENSION   A(9),  B(8)

C     LOCAL DATA VALUES
      DATA             DIASTD,DIASTC,DIASTP/0.000002,0.0000002,0.000002/
      DATA             ANGLPH,OUTPUT/0.15000000000,0.0/

C     TRANSFER TO COMPUTATION SECTIONS
      IF( WHICH  ) 301, 401, 501
```



```
C     INITIALIZATION ENTRY POINT
      ENTRY RANGEI
      PSIANG = PSII
 101  PSIA   = ABS(PSIANG)
 102  IF( ANGLEH  -  PSIA  ) 103, 111, 111
 103  IF( PIEO2L  -  PSIA  ) 104, 121, 121
 104  IF( PIEO2H  -  PSIA  ) 105, 121, 121
 105  IF(  PIEL   -  PSIA  ) 106, 112, 131
 106  IF(  PIEH   -  PSIA  ) 107, 112, 112
 107  PSIANG = PSIANG  -   SIGN( PIET2, PSIANG )
      GO TO 101
 111  DOWNRI =  DELTAI
      THETAP =  THETAI
      ZETA   =  PIEO2
      WHICH  =  0.0
      IF( OUTPUT ) 161, 142, 161
 112  DOWNRI =  - DELTAI
      THETAP =     THETAI
      ZETA   =  - PIEO2
      WHICH  =     0.0
      IF( OUTPUT ) 162, 142, 162
 121  IF( ANGLEH - ABS(DELTAI)  ) 122, 123, 123
 122  DOWNRI =  PIEO2
      THETAP =  THETAI   -   SIGN( PIEO2, PSIANG )
      ZETA   =  DELTAI
      IF( OUTPUT ) 172, 141, 172
 123  DOWNRI =  0.0
      THETAP =  THETAI
      ZETA   =  0.0
      WHICH  =  - 1.0
      IF( OUTPUT ) 173, 142, 173
 131  IF( ANGLEH - ABS(DELTAI)  ) 132, 133, 133
 132  SINDEL =  SIN(DELTAI)
      COSDEL =  COS(DELTAI)
      TANDEL =  SINDEL  /  COSDEL
      COSPSI =  COS(PSIANG)
      DOWNRI =  ATN(DDE  TANDEL  /  COSPSI  )
      THETAP =  ATN(DDE  ABS( SIN(PSIANG) ) / COSPSI * SINDEL  )
      ZETA   =  ATAN(   TANDEL  /  SIN(THETAP)    )
      THETAP =  THETAI   -   SIGN( THETAP, PSIANG )
      IF( OUTPUT ) 182, 141, 182
 133  DOWNRI =  0.0
      THETAP =  THETAI
      ZETA   =  PIEO2   -   ABS(PSIANG)
      IF( OUTPUT ) 183, 141, 183
 141  WHICH  =     1.0
 142  PSIP   =  SIGN( PIEO2, PSIANG)
      TPSIP  =  2.0  *  PSIP
      ONESPP =  SIGN( 1.0, PSIANG )
      ONESZE =  SIGN( 1.0, ZETA )
      PSIDEL =  0.0
      ILOOP  =  0
      ILOOPT =  0
      OUTPUT =  AINT( OUTPUT * 0.5 ) * 2.0
      GO TO 410
```



```
 151  A(1)   =  BRANCH
      A(2)   =  WHICH
      A(3)   =  PLACE
      A(4)   =  DELTAI   *   DEGRAD
      A(5)   =  THETAI   *   DEGRAD
      A(6)   =  PSII     *   DEGRAD
      A(7)   =  DOWNRI   *   DEGRAD
      A(8)   =  THETAP   *   DEGRAD
      A(9)   =  ZETA     *   DEGRAD

  11  FORMAT(  1H0, 3F9.0, 2X, 6F15.9  )
  12  FORMAT(             30X, 6015    )
  13  FORMAT(  6F15.9  )
  14  FORMAT(  6015    )

  91  FORMAT(  1H0, 3F9.0, 2X, 6F15.5  /   1H    )
  92  FORMAT(  1H ,         29X, 6015   /   1H    )

      WRITE(6,91)  A
      IF(-OUTPUT ) 152, 142, 142
 152  WRITE(6,92)  DELTAI,THETAI,PSII, DOWNRI, THETAP, ZETA
      GO TO 142

 161  PLACE  =  111.0
      GO TO 151
 162  PLACE  =  112.0
      GO TO 151
 172  PLACE  =  122.0
      WHICH  =    1.0
      GO TO 151
 173  PLACE  =  123.0
      GO TO 151
 182  PLACE  =  132.0
      WHICH  =    1.0
      GO TO 151
 183  PLACE  =  133.0
      WHICH  =    1.0
      GO TO 151
```



```
C     ALTERNATE ENTRY POINT WHICH DOES NOT REQUIRE A KICK-OFF
      ENTRY RANGEN

 291  IF( WHICH  ) 301, 401, 501

 301  TRDWNP =  ONESPP  *  ( THETA - THETAP )
      TCRSSR =  ONESPP  *  DELTA
      TPSIPG =  PSI     -  PSIP
      IF( OUTPUT ) 631, 401, 631

 401  DIFTHP =  ONESPP  *  ( THETA - THETAP )
 402  IF( -ANGLEH - DIFTHE ) 403, 421, 512
 403  IF(  ANGLEH - DIFTHE ) 404, 421, 421
 404  IF(  PIEO2L - DIFTHE ) 405, 531, 541
 405  IF(  PIEO2H - DIFTHE ) 406, 531, 531
 406  IF(   PIEL  - DIFTHE ) 407, 451, 541
 407  IF(   PIEH  - DIFTHE ) 408, 451, 451
 408  DOWNRI =  DOWNRI  -  PIE
      THETAP =  THETAP  +  TPSIP
      ZETA   =  - ZETA
      ONESZE =  -  ONESZE
      IF( WHICH  ) 301, 401, 501

 421  TRDWNR =  ONESZE  *  DELTA   -   DOWNRI
      TCRSSR =  0.0
 422  TPSIPG =  PSI    +   ( ONESZE - 1.0 )  *  SIGN( PIEO2, PSI )
      IF( OUTPUT ) 622, 401, 622

 451  TRDWNR =  PIE  -  ONESZE  *  DELTA   -   DOWNRI
      TCRSSR =  0.0
 452  TPSIPG =  PSI    -   ( ONESZE + 1.0 )  *  SIGN( PIEO2, PSI )
      IF( OUTPUT ) 652, 601, 652

 464  ONOSCD =  SIGN( 1.0, DIFTHE-PIEO2 )   +   1.0
      SNPPZE =  - ONESPP  *  ONESZE
      TDNDWR =  ONOSCD  *  PIEO2
      TCRSSR =  SIGN( TRDWND-DIFTHE, SNPPZE )
      TRDWNR =  TDNDWR    -    DOWNRI
      IF( ONOSCD ) 452, 422, 452
```

49


```
501  DIFTHE =  ( THETA - THETAP )  *  ONESPP
502  IF( -ANGLEN - DIFTHE ) 503, 521, 521
503  IF(  ANGLEN - DIFTHE ) 504, 521, 521
504  IF(  PIE02L - DIFTHE ) 505, 531, 541
505  IF(  PIE02H - DIFTHE ) 506, 531, 531
506  IF(   PIEL  - DIFTHE ) 507, 551, 541
507  IF(   PIEH  - DIFTHE ) 400, 551, 551

511  ZETA   =  SIGN( PIE, ZETA )  -  ZETA
512  PSIP   =  -   PSIP
     TPSIP  =  -  TPSIP
     ONESPP =  - ONESPP
     GO TO ( 501, 401, 501 ) WHICH

521  IF( ANGLEN - ABS(DELTA)    ) 522, 523, 523
522  TANHYP =  ABS(  SINDEL  ) / COSDEL
     ANGLE2 =  0.0
     ANGLE3 =  SIGN( PIE02, DELTA )   -   ZETA
     ANG3A  =  ABS(ANGLE3)
     GO TO 543
523  TDOWNW =  - DOWNWI
     TCRSSA =     0.0
     TPSING =     PSI    +   ONESPP  *  ( ZETA - PIE02 )
     IF( OUTPUT ) 623, 601, 623

531  TDOWNW =  DIFTHE    -    DOWNWI
     TCRSSA =  ONESPP  *  ( DELTA  -  ZETA )
     TPSING =  PSI    -    PSIP
     IF( PIE02H - ABS(TCRSSA)  ) 532, 534, 534
532  SPPTDE =  ONESPP  *  DELTA
     TDOWNW =  TDOWNW    *   PIE
     TCRSSA =  SIGN( PIE - ABS(TCRSSA), TCRSSA )
     TPSING =  TPSING    +   SIGN( PIE, SPPTDE )
     IF( OUTPUT ) 633, 601, 633
534  IF( OUTPUT ) 632, 603, 632

541  IF( ANGLEN - ABS(DELTA)    ) 542, 561, 561
542  SINDTH =  SIN(DIFTHE)
     COSDTH =  COS(DIFTHE)
     TANDTH =  SINDTH  /  COSDTH
     SINDEL =  SICDEL
     ANGLE1 =  ATAN(  SICDEL  /  COSDEL  /  SINDTH  )
     ANGLE2 =  ATN100(  TANDTH / ABS(SINDEL)  )
     TANHYP =  TANDTH / COS(ANGLE1)
     ANGLE3 =  ANGLE1    -    ZETA
     ANG3A  =  ABS(ANGLE3)
     IF(  ANGLEN - ANG3A   ) 543, 571, 571
543  IF(  PIE02L - ANG3A   ) 544, 572, 501
544  IF(  PIE02H - ANG3A   ) 545, 572, 572
545  IF(   PIEL  - ANG3A   ) 546, 573, 502
546  IF(   PIEH  - ANG3A   ) 502, 573, 573
```



```
551  IF( ANGLEN - ABS(DELTA)    ) 552, 553, 553
552  TANHYP =  - ABS( SINDEL )  /  COSDEL
     ANGLE2 =  PIE02
     ANGLE3 =  SIGN( PIE02, DELTA )   -   ZETA
     ANG3A  =  ABS(ANGLE3)
     GO TO 565
553  TDOWNW =  PIE    -    DOWNWI
     TCRSSA =  0.0
     TPSING =  PSI    -    ONESPP  *  ( ZETA + PIE02 )
     IF( OUTPUT ) 653, 601, 653

561  IF(  PIE02L - ABS(ZETA)    ) 562, 564, 565
562  IF(  PIE02H - ABS(ZETA)    ) 563, 564, 564
563  SINDTH =  SIN(DIFTHE)
     COSDTH =  COS(DIFTHE)
     TANHYP =  SINDTH  /  COSDTH
     SINAN3 =  SIN( - ZETA )
     COSAN3 =  COS( - ZETA )
     OSCOND =  SIGN( 1.0, COSDTH )   -   1.0
     PPZECD =  ONESPP  *  ONESZE  *  COSDTH
     OSZCND =  SIGN( 1.0, ONESZE*COSDTH )   -   1.0
     TDOWNW =  ATAN(  COSAN3  *  TANHYP  )   -   OSCOND  *  PIE02
     TCRSSA =  ATAN(  SIN(TDOWNW)  *  SINAN3  /  COSAN3  )  *  ONESPP
     ANGLE4 =  ATAN(  COSAN3  /  SIN(TCRSSA)  *  TANHYP  )
     TDOWNW =  TDOWNW    -    DOWNWI
     TPSING =  PST    +   SIGN( ANGLE4, PPZECD )   +   OSZCND  *  PSIP
     IF( OUTPUT ) 663, 601, 663
565  SINDTH =  SIN(DIFTHE)
     COSDTH =  COS(DIFTHE)
     TANHYP =  SINDTH  /  COSDTH
     SINAN3 =  SIN( - ZETA )
     COSAN3 =  COS( - ZETA )
     OSZEND =  ONESZE    -    1.0
     TDOWNW =  ATN100(  COSAN3  *  TANHYP  )
     TCRSSA =  ATAN(  SIN(TDOWNW)  *  SINAN3  /  COSAN3  )  *  ONESPP
     ANGLE4 =  ATN100(  COSAN3  /  ABS( SIN(TCRSSA) )  *  TANHYP  )
     TDOWNW =  TDOWNW    -    DOWNWI
     TPSING =  PST    +   SIGN( ANGLE4, TCRSSA )   +   OSZEND  *  PSIP
     IF( OUTPUT ) 665, 601, 665
```



```
571  SPPDDE =  - ONESPP  *  DELTA
     OSZEND =  ONESZE    -    1.0
     TDOWNW =  ATN100(  TANHYP  )   -   DOWNWI
     TCRSSA =  0.0
     TPSING =  PSI    +   SIGN( ANGLE2, SPPDDE )   +   OSZEND  *  PSIP
     IF( OUTPUT ) 671, 601, 671
572  OSCOND =  SIGN( 1.0, COSDTH )   -   1.0
     SPPDZE =  - ONESPP  *  ONESZE
     SPPDDE =  - ONESPP  *  DELTA
     SOZCND =  SIGN( 1.0, DELTA ) + SIGN( 1.0, ONESZE*COSDTH ) - 1.0
     TDOWNW =  - OSCOND  *  PIE02
     TCRSSA =  SIGN( TDOWNW - ATN100(TANHYP), SPPDZE )
     TPSING =  PSI    +   SIGN( ANGLE2, SPPDDE )   +   SOZCND  *  PSIP
     IF( OUTPUT ) 672, 601, 672
573  OSCOND =  SIGN( 1.0, COSDTH )   -   1.0
     SPPDDE =  - ONESPP  *  DELTA
     SOZCND =  SIGN( 1.0, DELTA ) + SIGN( 2.0, ONESZE*COSDTH ) - 1.0
     TDOWNW =  ATAN(  - TANHYP  )   -   OSCOND  *  PIE02
     TCRSSA =  0.0
     TPSING =  PSI    +   SIGN( ANGLE2, SPPDDE )   +   SOZCND  *  PSIP
     IF( OUTPUT ) 673, 601, 673

581  SPPDDE =  - ONESPP  *  DELTA
     SOASND =  ( 1.0 - SIGN( 1.0, DELTA*ANGLE3 ) )  ) * ONESZE   -   1.0
     SINAN3 =  SIN(ANGLE3)
     COSAN3 =  COS(ANGLE3)
     TDOWNW =  ATN100(  COSAN3  *  TANHYP  )
     TCRSSA =  ATAN(  SIN(TDOWNW)  *  SINAN3  /  COSAN3  )  *  ONESPP
     ANGLE4 =  ATN100(  COSAN3  /  ABS( SIN(TCRSSA) )  *  TANHYP  )
     TDOWNW =  TDOWNW    -    DOWNWI
     TPSING =  PSI    +   SIGN( ANGLE2, SPPDDE )
    1                 +   SIGN( ANGLE4, TCRSSA )   +   SOASND  *  PSIP
     IF( OUTPUT ) 681, 601, 681
582  OSCOND =  SIGN( 1.0, COSDTH )   -   1.0
     SPPDDE =  - ONESPP  *  DELTA
     OSSA3A =  SIGN( 1.0, PIE - ABS(ANGLE3)  )
     SAAPZC =  OSSA3A  *  ONESPP  *  ONESZE  *  COSDTH
     SOZCND =  SIGN(1.0,DELTA) + SIGN(2.0-OSSA3A,ONESZE*COSDTH) - 1.0
     SINAN3 =  SIN(ANGLE3)
     COSAN3 =  COS(ANGLE3)
     TDOWNW =  ATAN(  COSAN3  *  TANHYP  )   -   OSCOND  *  PIE02
     TCRSSA =  ATAN(  SIN(TDOWNW)  *  SINAN3  /  COSAN3  )  *  ONESPP
     ANGLE4 =  ATAN(  COSAN3  /  SIN(TCRSSA)  *  TANHYP  )
     TDOWNW =  TDOWNW    -    DOWNWI
     TPSING =  PSI    +   SIGN( ANGLE2, SPPDDE )
    1                 +   SIGN( ANGLE4, SAAPZC )   +   SOZCND  *  PSIP
     IF( OUTPUT ) 682, 601, 682
```



```
691  IF(  ABS(TDOWNW)   -  STASTD   ) 602, 693, 693
692  TDOWNW =  0.0
693  IF(  ABS(TCRSSA)   -  STASTC   ) 694, 695, 695
694  TCRSSA =  0.0
695  IF(  PIE02L -   ABS(TPSING-PSINGL)  ) 696, 697, 697
696  TPSING =  TPSING    +    SIGN( PIE02, PSINGL-TPSING )
     ILOOP  =  ILOOP    +    1
     IF( ILOOP  .LT. 100  ) GO TO 695
     ILOOP  =  0
     WRITE(6,12)  DELTA, THETA, PSI,  DOWNWI, THETAP, ZETA
     WRITE(6,14)  DIFTHE, TDOWNW, TCRSSA, TPSING,
    1             ANGLE1, ANGLE2, ANGLE3, ANGLE4
     ILOMPT =  ILOMPT   +    1
     IF( ILOMPT .EQ.  10  ) ILAST = 1
     IF( OUTPUT .EQ. 0.0  ) OUTPUT = 1.0
697  IF(  ABS(TPSING)   -  STASTP   ) 698, 699, 699
698  TPSING =  0.0
699  PSINGL =  TPSING

     DWNWA  =  TDOWNW  *  DGTRAD
     CROSSA =  TCRSSA  *  DGTRAD
     PSING  =  TPSING  *  DGTRAD

610  RETURN
```

```
  611 A(1)  = BRANCH
      A(2)  = GROUPP
      A(3)  = PLACE
      A(4)  = DELTA   = [illegible]
      A(5)  = THETA   = [illegible]
      A(6)  = PSI     = [illegible]
      A(7)  = SOUND1  = [illegible]
      A(8)  = THETAP  = [illegible]
      A(9)  = ZETA    = [illegible]
      B(1)  = DEPTHE  = [illegible]
      B(2)  = TORQUE  = [illegible]
      B(3)  = TCRSOR  = [illegible]
      B(4)  = TPSING  = [illegible]
      B(5)  = ANGLE1  = [illegible]
      B(6)  = ANGLE2  = [illegible]
      B(7)  = ANGLE3  = [illegible]
      B(8)  = ANGLE4  = [illegible]
      WRITE(6,11) A
      IF(-OUTPUT ) 612, 613, 613
  612 WRITE(6,12) DELTA, THETA, PSI,  SOUND1, THETAP, ZETA
  613 WRITE(6,13) B
      IF(-OUTPUT ) 614, 601, 601
  614 WRITE(6,16) DEPTHE, TORQUE, TCRSOR, TPSING,
     1              ANGLE1, ANGLE2, ANGLE3, ANGLE4
      GO TO 601
  627 PLACE  =  627.0
      GO TO 611
  623 PLACE  =  623.0
      GO TO 611
  631 PLACE  =  631.0
      GO TO 611
  632 PLACE  =  632.0
      GO TO 611
  633 PLACE  =  633.0
      GO TO 611
  642 PLACE  =  642.0
      GO TO 611
  651 PLACE  =  651.0
      GO TO 611
  663 PLACE  =  663.0
      GO TO 611
  665 PLACE  =  665.0
      GO TO 611
  671 PLACE  =  671.0
      GO TO 611
  672 PLACE  =  672.0
      GO TO 611
  673 PLACE  =  673.0
      GO TO 611
  681 PLACE  =  681.0
      GO TO 611
  682 PLACE  =  682.0
      GO TO 611

      END
```



```
$IBMAP ACN100  35,00,DECK
*
*       THIS PROGRAM IS DESIGNED TO CONVERT THE FUNCTION SUBROUTINE ATAN
*       FROM THE RANGE  -  ( - 90 DEGREES  TO  + 90 DEGREES )        VERSION 6.0
*         TO THE RANGE  -  ( + 0 DEGREES   TO  + 180 DEGREES )
        LBL     ACN100,BEGIN,
        SPACE   2
        ENTRY   ATN100
ATN100  CLA     3,4             FIND THE VALUE OF THE ARGUMENT
        TZE     ZERO            TRANSFER TO ZERO IF ARGUMENT IS + OR - ZERO
        STO     ANGLE           SET UP THE ARGUMENT FOR THE ARCTANGENT
        SXA     SYSLOC,4        UPDATE SYSLOC ADDRESS
        STD     SYSLOC,0        UPDATE SYSLOC DECREMENT
        SXA     LK,BP,4         UPDATE LINKAGE DIRECTOR
        SXA     SAVE4,4         SAVE INDEX REGISTER 4
        TSX     ATAN,4          COMPUTE THE VALUE OF THE ARCTANGENT
        TXI     *+3,,1
        PZE     1,,LK,BR
        PZE     ANGLE
SAVE4   AXT     **,4            RESTORE INDEX REGISTER 4
        STD     SYSLOC,4        UPDATE SYSLOC DECREMENT
        TPL     1,4             IF ARGUMENT IS POSITIVE, RETURN
        FAD     PIE             IF ARGUMENT IS NEGATIVE, ADD PIE
        TRA     1,4             RETURN
ZERO    TPL     1,4             IF ARGUMENT IS + ZERO, RETURN
        CLA     PIE             IF ARGUMENT IS - ZERO, REPLACE BY PIE
        TRA     ,1,4            RETURN
        SPACE   2
*       CONSTANTS AND STORAGE
PIE     DEC     3.14159265
ANGLE   BSS     1,0
        SPACE   2
LK,BP   DEC     0
        BCI     1,ACN100
        SPACE   10
        END
```

```
STATIC RCDI    PULIST,PMAP,RD,DECK
      SUBROUTINE RADIUS

C     SUBROUTINE RADIUS CODE NUMBER RCDIUS                 VERSION 4.0

C     THIS PROGRAM WILL COMPUTE THE GEOCENTRIC RADIUS FOR A GIVEN
C     HEIGHT, LATITUDE, AND LONGITUDE OVER A TRIAXIAL ELLIPTIC MODEL
C     OF THE EARTH.

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CVAR/    VDUMP(4),RD,VDUMSP(4),DELTAD,THETAD,VDUMB(30)
      COMMON /CIVAR/   IVDUMP(2),VINITD,IVDUMM(4),DINIT,TOINIT
      COMMON /CPIES/   PDUMP(4),RZERO,PDUMB(10)

C     DOUBLE PRECISION STATEMENTS FOR GLOBAL VARIABLES
      DOUBLE PRECISION RD,DELTAD,THETAD
      DOUBLE PRECISION VINITD,DINIT,TOINIT
      DOUBLE PRECISION RZERO

C     EQUIVALENCE STATEMENTS FOR GLOBAL VARIABLES
      EQUIVALENCE      (R,RD),(DELTA,DELTAD),(THETA,THETAD)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CRADUS/  BSHAPE,GSHAPE,REQUD,REQU2D,FYD1,FYD2,FYD3,FYD4,
     1                 FYS03,FYS04,FYS07,FYS08,FYS09,FEARTH,EEARTH,
     2                 THETA2,RZERO,RZ2D
C     DOUBLE PRECISION STATEMENTS FOR SUBROUTINE VARIABLES
      DOUBLE PRECISION REQUD,REQU2D,FYD1,FYD2,FYD3,FYD4,RZEROD,RZ2D

C     EQUIVALENCE STATEMENTS FOR SUBROUTINE VARIABLES
      EQUIVALENCE      (FYS03,FYD2),(FYS03,FYD3),(FYS04,FYD4)

C     MISCELLANEOUS TYPE AND DIMENSION STATEMENTS FOR LOCAL VARIABLES
      DOUBLE PRECISION FYD,DFYD
```



```
C     PRELIMINARY COMPUTATION
      IF(   ABS(BSHAPE)   ) 301, 151, 101

 101  RD     =  REQUD   +   VINITD
      DELTAD =  DINIT   +   RZERO
      THETAD =  TOINIT  +   RZERO
      KOUNT  =  0
      COSD   =  COS(DELTA)
 102  IF(-BSHAPE ) 103, 103, 104
 103  SIND   =  SIN(DELTA)
      SIND2  =  SIND**2   -   0.5
      STCOL2 =  ( SIND * COSD ) ** 2
      FYS02  =  FYS09  *  SIND**2
      IF( BSHAPE ) 201, 104, 201
 104  FYS04  =  FYS09*(( SIN(THETA - THETA2)**2 - 0.5 ))*COSD
      GO TO 201

 151  R      =  RZERO   +   VINITD
      GO TO 301

C     ACTUAL COMPUTATION OF THE GEOCENTRIC RADIUS
 201  FYD    =  RD      +   FYD1
 202  IF(-BSHAPE ) 203, 203, 206
 203  FYS03  =  (( FYS06 / R  +  FYS07  +  FYS08 * SIND2 )) * STCOL2
      FYD    =  FYD     +   FYD2   +   FYD3
      IF( BSHAPE ) 211, 206, 211
 206  FYD    =  FYD     +   FYD4

 211  DFYD   =  FYD     -   VINITD
      IF( 1.0E-07  -   ABS(DFYD)    ) 212, 301, 301
 212  RD     =  RD      -   DFYD
      KOUNT  =  KOUNT   +   1
      IF( 10 -  KOUNT ) 301, 201, 201

 301  RETURN

      END
```

SUBROUTINE PCTCHI 6.0   PAGE   1

```
$IBFTC PCHI    FULIST,PMAP,DD,DECK
      SUBROUTINE PITCHI

C     SUBROUTINE PITCHI CODE NUMBER PCTCHI                    VERSION 6.0

C     THIS PROGRAM WILL INITILIZE THE ATTITUDE OF THE VEHICLE SO AS TO
C     MINIMIZE TEMPERATURE AND ACCELERATION PEAKS AND MAXIMIZE CROSS
C     RANGE.

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CONTRL/  SMSSNT,TNCNTS,GOPTNS,DDUMB(10)
      COMMON /CFVAR/   FV2,FDUMV2,FV,FDUMVB(3),FY,FDUMM(11),FTI,FDUMG(7)
      COMMON /CSCAP/   ALPHAD,ALPOLD,SINALP,COSALP,PHIDEG,PHIOLD,SINPHI,
     1                 COSPHI
      COMMON /CINTS/   M,N,NTP,IGPDVD,IGNORE,KOUNT,INTLST,ILAST,JFAIL,
     1                 IDUMA(11)
      COMMON /CPIFS/   PDUMF(2),PIE02,PDUMH(7),RDDDGA,DGORDR,PDUMD(6)

C     COMMON STATEMENTS FOR GOPTNS VARIABLES
      COMMON /CNIOS/   DICNDF(3),TCEXTS,DIDDOM(77),ITHOOS(9),DIODDN(5),
     1                 ITHORM,ITHOBL,DI999A(4)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CDBSET/  ANGINC,ATALIC,ATPHIC,DSDUMB(7)
      COMMON /COPNV/   BRDUMF(19),GREFRZ,TGNFRZ
      COMMON /CPITCH/  ATTACK,ATARNF,ATKALP,ATKPHI,BANK,CPSMAX,ROLL,
     1                 COSFM,CAD02N,CL1C2N,HDD,DRAG,HIPT,HIPTSP,HIPTCP,
     2                 PSIRG ,PSIRGC,CSCPOR,ALPRAT,PHIRAT,PHIRTE,BIASPD

C     COMMON STATEMENTS FOR LOCAL VARIABLES
      COMMON /CPITVR/  PTINIT,KONTOL,ROTCNG,PHICNG,JOS,CDZERO,CP,CPDRAG,
     1   CLPHAU,COSONU,COS2NU,CLS2NU,ALPHAL,COSONL,COS2NL,CLS2NL,PHIF,
     2   ALPHAF,COSONF,COS2NF,CLS2NF,PHIU,PHIULT,PHIL,SINPHL,COSPHL,
     3   PHILPT,PHILMN,PHILDF,FVTMAT,FYTMAX,FEGTMX,ICPSNX,ICESVE,
     4   FVMIN,FYMIP,FEGMIN,FVNIFF,FYDIFF,FEDDIF,PI2SCH,SINPHF,COSPHF,
     5   GNMEAL,GAINAL,ALPSTR,ALPNEW,ALPDAT,ALPONE,ALPOST,PTIANE,
     6   GNMZPH,GAINPH,PHISTR,PHINE ,PHIDAT,PHIONE,PHIOST,PTIPHE,PHISTE
C     DIMENSION STATEMENTS FOR LOCAL VARIABLES
      DIMENSION        ALPSVE(100),PHISVE(100),TIMSVE(100),ROT(9)
```

SUBROUTINE PCTCHI 6.0   PAGE   2

```
C     FORMATS
   10 FORMAT(  7F10.0/7F10.0/7F10.0/7F10.0/7F10.0/2F10.0,5I10/7F10.0   )
   11 FORMAT(  7F10.0, I2  )
   13 FORMAT(  11GHO   NOTE -   IN ORDER TO MINIMIZE THE TEMPERATURE AND
     1 ACCELERATION PEAKS, THE ATTITUDE OF THE VEHICLE WILL BE SLOWLY  /
     2 13X, 101HCHANGED TO THE FOLLOWING VALUES IN PLACE OF THE NOMINAL
     3 (OR TERMINAL) VALUES GIVEN ABOVE WHENEVER THE                   /
     4 13X, 101HEQUILIBRIUM STAGNATION POINT TEMPERATURE IS INCREASING.
     5   THE ATTITUDE WILL SLOWLY CHANGE BACK TO THE                   /
     6 13X, 107HNOMINAL VALUES WHENVER THE TEMPERATURE IS DECREASING.
     7 THE FOLLOWING RATES OF CHANGE WILL BE USED  -                   /
     8 13X, 20HMAXIMUM PITCH RATE =, F8.3, 30H DEGREES/SEC.      MAXIMUM
     9 ROLL RATE =, F8.3, 13H DEGREES/SEC.                             /
     A 13X, 20HMINIMUM PITCH RATE =, F8.3, 30H DEGREES/SEC.      MINIMUM
     B ROLL RATE =, F8.3, 13H DEGREES/SEC.                             )
   14 FORMAT(  12HO   PHI     =, F9.3, 19H DEGREES   CD S/M =, F11.7, 20H
     1 SQ.FT./SLUG   S/D =F11.7,20H   ANGLE OF ATTACK =F9.5,8H DEGREES /
     2 17X 20HCOEFFICIENTS -        LIFT =, F11.7, 14X 6HDRAG =, F11.7 )
   15 FORMAT(  13X, 20HUPPER PITCH ACCEL. =, F8.3, 30H DEGREES/SEC**2
     1 UPPER ROLL ACCEL. =, F8.3, 19H DEGREES/SEC**2                   /
     2 13X,    20HLOWER PITCH ACCEL. =, F8.3, 30H DEGREES/SEC**2    LOWER
     3 ROLL ACCEL. =, F8.3, 19H DEGREES/SEC**2                          )
   17 FORMAT(     115HO   NOTE -   IN ORDER TO MAXIMIZE CROSS RANGE, THE
     1 NOMINAL ANGLE OF ROLL (NOMINAL MAXIMUM IF TEMPERATURES ARE BEING/
     2 13X, 30HMINIMIZED) WILL VARY (ACCORDING TO THE, F4.1,   13H POWER
     3 OF THE, A6, A6, 8H BETWEEN, F9.3, 8H DEGREES / 13X, 3HAND, F9.3,
     4 29H DEGREES OVER A SPECIFIED, A6, A3, 7H RANGE.                  )
   18 FORMAT(  119HO   NOTE -   IN ORDER TO EXTEND (OR SHORTEN) THE DOWN
     1 RANGE, THE FOLLOWING PAIRS OF LIMITS WILL BE USED FOR THE ANGLES/
     2 13X,101HOF ATTACK AND ROLL UNTIL THE EQUILIBRIUM STAGNATION POINT
     3 TEMPERATURE STARTS ITS SECOND RISE (OR SOME                     /
     4 13X, 94HOTHER APPROPIATE CONDITION IS REACHED IN THE FLIGHT) -  )
   71 FORMAT(  9H ALPOST =F8.3, 11H   PTIAND =F8.3, 11H   ALPHID =F8.3,
     1 11H   PTIANE =F8.3, 11H   ALPNEW =F8.3, 10H   NOALP =, I8        )
   72 FORMAT(  9H PHIOST =F8.3, 11H   PTIPHD =F8.3, 11H   PHIMID =F8.3,
     1 11H   PTIPHE =F8.3, 11H   PHINEW =F8.3, 10H   NOPHI =, I8        )
   74 FORMAT(  9HAVRAGE =F8.3, 11H   TIMDIF =F8.3, 11H   ALPDAT =F8.3,
     1 11H   PHIDAT =F8.3, 11H   ATKSVE =F8.3, 10H   ND    =, 2I8       )
   99 FORMAT(  15HO      (ERROR =, A115  /  11F11.4  /  8O19  )
```

SUBROUTINE PCTCHI 6.0   PAGE   3

```
C     LOCAL DATA VALUES UNDER CONTROL OF GOPTNS 2000
      DATA    DALPHS,DALPTS,ALPHAI,NODI,BAOL,CRNROL,ROLD,PI2SOL/
     1        1.0,1.073741823E9,70.0,-0.5,1.0,1.0,1.0,1000.0/
      DATA    PRTOL1,PRTOL2,PRGCOL,CSCPOL,ZEROM/
     1        0.785398164,0.392699082,1.57079633,0.0,-0.0/
      DATA    RTEADJ,GAINOL,RATNPY,NORSVE/1.0,10.0,1.0,0/
      DATA    FVMAX,FVMNDF/10000.0,500.0/
      DATA    FYMAX,FYMNDF/1000000.0,10000.0/
      DATA    FEGMAX,FEGMND/100000000.,1000000.0/
      DATA    ALPONI,ALPON2,ALPONE,ALPOS/2.05,0.5,0.125,1.3/
      DATA    NOSUB,NOMAX,MULTND,MINPRC,MAXPRC,TIMMIN/3010,110,310,15.0/
      DATA    ALPSUM,PHISUM,GNMIAL,GNANPH/0.75,1.5,1.0,2.0/

C     LOCAL DATA VALUES UNDER CONTROL OF GOPTNS 4000
      DATA    ALPHAE,NODE,ALPHAS,NODS,PHISKP/70.0,10.0,-0.0,-22.5/
      DATA    ICRSOL/1/

C     LOCAL DATA VALUES WHICH ARE FIXED
      DATA    RGT(4)/30H VELOCITY).  ALTITUDE).   ENERGY ). /
      DATA    O1,O0377,O2000,O4000/O1,O377000000000,O2000,O4000/
      DATA    ON60,ON100/O40000000,O100000000/

C     INITILIZATION OF CONTROL VARIABLES
      ROUNDS =  3.0
      ALPSVE =  - COSOM
      ALPOLD =  5000.0
      PHISVE =  PHIDEG
      PHIOLD =  2000.0
      IPPD   =  0
      NOTOT  =  0
      NOMLST =  0.0
      OUTTST =  AMOD( COSOM, 1000.0 )
      TEST   =  HOD
      PTINIT =  7.5E29
      PHICNG =  0.0
      KONTOL =  - 1
      ROTCNG =  - 1
```

SUBROUTINE PCTCHI 6.0   PAGE   4

```
      IF( NORSVE )  81,  80,  81
   80 ALPRSV =  ALPRAT
      PHIRSV =  PHIRAT
      NORSVE =  1
      HOS    =  40.0
      CDZERO =  0.0625
      CP     =  1.75
      CPDRAG =  1.4975
      FVMIN  =     5000.0
      FYMIN  =   100000.0
      FEGMIN =  10000000.0
      FVTMAX =     24000.0
      FYTMAX =   250000.0
      FEGTMX =  90000000.0
   81 IF( AND( GOPTNS, O2000  ) )  82,  83,  82
   82 READ (5,10)  GREFRZ,HOS,CDZERO,CP,CPDRAG, DALPHS,DALPTS,
     1             ALPRSV,PHIRSV,PHIRTE,RAOL,CRNROL,PI2SOL,ROLD,
     2             ALPHAI,NODI,PRTOL1,PRTOL2,PRGCOL,CSCPOL,RTEADJ,
     3             FVMIN,FVMAX,FVMNDF,FYMIN,FYMAX,FYMNDF,GAINOL,
     4             FEGMIN,FEGMAX,FEGMND,ALPONI,ALPON2,ALPONE,ALPOS,
     5             RATNPY,TIMMIN,NOSUB,NOMAX,MULTND,MINPRC,MAXPRC,
     6             ALPSUM,PHISUM,GNMIAL,GNANPH,FVTMAX,FYTMAX,FEGTMX
   83 IF( AND( GOPTNS, O4000  ) )  84,  85,  84
   84 READ (5,11)  ALPHAE,NODE,ALPHAS,NODS,PHISKP,ALPHAI,NODI,ICRSOL
   85 IF( CPDRAG )  86,  86,  87
   86 CPDRAG =  CP  -  CDZERO
   87 TGNFRZ =  GREFRZ * 2.0
      ALPRAT =  ALPRSV
      PHIRAT =  PHIRSV
      ALPONE =  0.0
      PHIONE =  0.0
      ALPDAT =  ALPRSV * DGORDR
      ALPONE =  ALPDAT
      ALPONN =  ALPONI
      ALPON2 =  ALPONN * 2.0
      AVRAGE =  0.0
      TIMDIF =  0.0
      GNMIAL =  AMAX1( GAINOL, 0.0 )
      NOSUB  =  ( MAX0( NOSUB, 4 ) / 2 ) * 2
      IF( GNMIAL .EQ.  0.0 ) NOSUB = ( MAX0( NOSUB, 6 ) / 2 ) * 2
      RATFAC =  AMAX1( AMOD( ABS( RATNPY ), 100.0 ), 1.0 )
      IF( -RATNPY .GT. 1.0 ) RATFAC = 1.0 / RATFAC
   89 DALPHA =  DALPHS

   91 IF(   AMOD(OUTTST)   )  92,  93,  92
   92 IF(         OUTTST   )  93,  94,  93
   93 ALPHAD =  - OUTTST
      GO TO 101
   94 ALPHAD =  90.0
```

33


```
101  CALL SNCSAL( 81011 )
     ALPHAR =  ALPHAD * RDNDGR
     SINALP =  SIN( ALPHAR )
     COSALP =  COS( ALPHAR )
1011 CONTINUE
     CL     =  ABS( SINALP ) * SINALP
     CD     =  CDZERO   +   CPOPAG * CL * SINALP
     CL     =  CP * CL * COSALP
     HOD    =  CL / CD
     COSON  =  CD * GAFFRZ / WOS
     COSO2N =  COSON / 2.0
     CLSO2N =  COSO2N * HOD
     PHIDEG =  SIGN( AMIN1( AMOD(ABS(PHIDEG),900.0), 90.0), PHIDEG )
     CALL SWFSPH( 81012 )
     PHIRAD =  PHIDEG * RDNDGR
     SINPHI =  SIN( PHIRAD )
     COSPHI =  COS( PHIRAD )
1012 CONTINUE
     IF(   BOOL( TEST )    ) 102, 109, 102
102  IF(         TEST      ) 111, 121, 111
103  IF(         OUTTST    ) 104, 105, 104
104  WRITE(6,14)  PHIDEG, COSON, HOD, ALPHAD, CL, CD
105  IF(   BOOL(THCNTS)    ) 106, 107, 106
106  IF(         GRUNDS    ) 151, 107, 151
107  RETURN

111  IF(  HOD  -  ABS(TEST)     ) 112, 103, 114
112  DALPRF =  DRE  AMO( ALPHAD, DRS77 ),  DI  )
     ALPHAD =  SIGN( AMAX1( DALPRF+0.0, DALPHA ), TEST )   +   ALPHAD
     IF( 90.0  -  ABS(ALPHAD)  ) 131, 101, 101
114  ALPHAD =  SIGN( DALPHA, - TEST )   +   ALPHAD
115  DALPHA =  DALPHA  *  0.5
116  IF( ALPHAD  /  DALPHA - DALPTS ) 101, 103, 103

121  IF( HOD    -  HODLST ) 122, 123, 124
122  ALPHAD =  ALPHAD   -   DALPHA
123  ALPHAD =  ALPHAD   -   DALPHA
     HODLST =  0.0
     GO TO 115
124  ALPHAD =  ALPHAD   +   DALPHA
     HODLST =  HOD
     GO TO 101

131  IERR   =  IERR  +   1
     IF( 2-IERR ) 132, 133,  88
132  ALPHAD =  0.0
     DALPHA =  DALPHS
     TEST   =  - 0.0
     GO TO 101
133  DALPHA =  DALPHS
     IF( - TEST ) 134, 1339, 96
1339 IERROR =  1339
     GO TO 999
134  ALPHAD =  0.0
     GO TO 101
```



```
151  BOUNDS =  BOUNDS  -  2.0
     IF( BOUNDS +  1.0    ) 191, 152, 161
152  ALPHAU =  ALPHAD
     COSOMU =  COSON
     COS2MU =  COSO2N
     CLS2MU =  CLSO2N
     PHIU   =  SIGN( 0.0, PHIDEG )
     PHIULT =  SIGN( 1.0, PHIDEG )
     IF( ALPHAL -  ALPHAU -  PI29CH ) 153, 155, 1529
1529 IERROR =  1529
     GO TO 999
153  WRITE(6,13)  ALPONE,PHIONE,ALPOWN,PHIOWN
     IF( GAMMAL .GT. 0.0  ) WRITE(6,15) GAMMAL,GAMXPH,GXMMAL,GMXXPH
154  WRITE(6,14)  PHIDEG, COSON, HOD, ALPHAD, CL, CD
     GO TO 156
155  THCNTS =  0.0
156  IF( CAMERG ) 157, 101, 157
157  WRITE(6,17)  RGT(1),RGT(2),RGT(3),PHIL,PHILMN,RGT(2),RGT(3)
     GO TO 101
161  ATTACK =  - 6.0
     EPTS   =  161.0
     ATKALP =  161.0
     ALPHAL =  ALPHAD
     COSOML =  COSON
     COS2ML =  COSO2N
     CLS2ML =  CLSO2N
     TEST   =  0.0
     OUTTST =  0.0
     IF( THCNTS ) 162, 163, 162
162  ALPHAD =  ALPHAT
     TEST   =  HODT
     DALPHA =  DALPHS
     IERR   =  0
     OUTTST =  SIGN( 0.0, TEST )
163  PHIDAT =  SIGN( PHIDSV * DGRDR, - PHIDEG )
     PHIONE =  PHIDAT
     ALPHRT =  ALPDAT / PHIDAT
     PHALRT =  PHIDAT / ALPDAT
     PHIDN1 =  PHALRT * ALPONE
     PHIDN2 =  PHALRT * ALPON2
     PHIOWN =  PHALRT * ALPOWN
     GAMXPH =  PHALRT * GAMMAL * ABS( RTEADJ )
     GAMXPH =  SIGN( GMAXPH, GAMXPH )
     PHIDA2 =  - ABS( PHION2 )
     PHIDA1 =  - ABS( PHION1 )
     IF( RTEADJ ) 1632,1631,1633
1631 GAMMAL =  0.0
     GAMXPH =  0.0
     NOSWD  =  ( MAX0( NOSWD, 6 ) / 2 ) * 2
     IF(   BOOL(RTEADJ)    ) 1633,1632,1633
1632 ALPOWN =  ALPONE
     PHIOWN =  PHIONE
1633 PHIDAE =  ABS( PHIONE )
     PHIDAW =  ABS( PHIOWN )
     GMAXPH =  ABS( GAMXPH )
```



```
164  PHIL   =  PHIDEG
     PHILDT =  PHIDEG
     SINPHL =  SINPHI
     COSPHL =  COSPHI
     PHIDTF =  SIGN( PHIPTF * DGRDR, - PHIDEG )
     BERG   =  BERL
     BANK   =  0.0
     PI29RG =  PI29RL * AMIN1( ABS(THCNTS), 1.0 )
     PI29CH =  AMIN1( PI29RG, 0.125 )
     CONRSV =  AMIN1( AINT( ABS(PHISVT) / 990.0), 9.0 ) * CONRDL
     ICRVRE =  CONRSV
     CONRDG =  0.0
     IF( CONRSV  .GT. 0.0 ) CONRDG = 2.0
     COSMAX =  ( SIGN( 1.0, 0.0625-PI29CH ) + 1.0 ) * CONRDG
     PSIRAT =  PRTRL1   +  PRTRL2 * AMIN1( CONRDG, 1.0 )
     PSISRC =  PRRPOL
     ZSCPHM =  CSCPOL
     ROLL   =  ABS( SIGN( 1.0, PI2R2 -PSIRAT )  + 1.0 ) * ARLD
     IF( ABS(PHILI - 90.0 ) 165, 168, 168
165  PHILMP =  PHIDEG
     PHILMN =  SIGN( 0.0, PHIDEG )
     IF( PHIDEG ) 166, 169, 166
166  IF( PI29CH ) 167, 171, 167
167  PHIDEG =  SIGN( 9.9, PHIDEG )
     PHIOLD =  PHIDEG
     SINPHI =  PHIDEG
     COSPHI =  1.0
     GO TO 171
168  PHILMP =  SIGN( 180.0, PHISVE )  -  AMOD( PHISVE, 900.0 )
     PHILMN =  SIGN(  90.0, PHISVE )  -  PHILMP
     IF( PI29CH ) 167, 171, 167
169  BERG   =  0.0
     PI29RG =  0.0
     PI29CH =  0.0
     CONRRG =  0.0
     COSMAX =  0.0
     ROLL   =  0.0
     GAMXPH =  0.0

171  IF( CONRRG ) 172, 101, 172
172  IF( CONRSV -  4.0    ) 173, 173, 174
173  RGT(1) =  CONRSV / 2.0
     RGT(2) =  RGT(4)
     RGT(3) =  RGT(9)
     GO TO 191
174  IF( CONRSV -  8.0    ) 175, 175, 176
175  RGT(1) =  ( CONRSV - 4.0 ) / 2.0
     RGT(2) =  RGT(6)
     RGT(3) =  RGT(7)
     GO TO 191
176  RGT(1) =  1.0
     RGT(2) =  RGT(9)
     RGT(3) =  RGT(9)
     GO TO 191
```



```
181  IF( ALPSVP.GT.1000.0 ) GO TO 184
182  IF( ICRSML.LT.1      ) GO TO 187
183  ICRSVP =  ICRSML
     ICPSML =  MIN0( ICRSML+10, 13 ) * MIN0( ICRSML, 1 )
     GO TO 188
184  ICRSVP =  ICRSML
     ICPSML =  MIN0( ALPSVP/1000.0+10.0,13.0 ) * MIN0( ICPSAV, 1 )
188  ALPHAP =  ALPHAL
     COSOMP =  COSOML
     COS2MP =  COS2ML
     CLS2MP =  CLS2ML
     PHIP   =  PHIL
     SINPHP =  SINPHL
     COSPHP =  COSPHL
186  IF( ICRSML.LT.10     ) GO TO 187
     PHIDEG =  SIGN( PHISRP, PHIDEG )
187  ALPHAD =  ALPHAS
     TEST   =  HODS
     DALPHA =  DALPHS
     OUTTST =  - ALPHAS
     IERR   =  0
189  WRITE(6,19)
     GO TO 101

191  ALPHAD =  ALPHAD
     COSOML =  COSON
     COS2ML =  COSO2N
     CLS2ML =  CLSO2N
     PHIL   =  PHIDEG
     PHILDT =  PHIDEG
     SINPHL =  SINPHI
     COSPHL =  COSPHI
192  ALPHAD =  ALPHAP
     TEST   =  HODP
     DALPHA =  DALPHS
     OUTTST =  - ALPHAD
     IERR   =  0
193  BOUNDS =  0.0
     IF( PI29CH ) 194, 101, 194
194  PHIDEG =  SIGN( 0.0, PHIDEG )
     GO TO 101
```

```
C     ADJUSTMENT OF ATTACK AND ATAKNE ENTRY POINT
      ENTRY PITCHA

      ATKSVE = ATTACK
 201  IF( N - 1 ) 2019,202, 2019
2019  IERROR = 2019
      GO TO 999
 202  IF( IOPOVD + 200     ) 2029,203, 2239
2029  IER  R = 2029
      GO    999
 203  IF( IONDPE ) 2039,204, 2039
2039  IERROR = 2039
      GO TO 999
 204  IF( EPTS  ) 2049,211, 205
2049  IERROR = 2049
      GO TO 999
 205  EPTS   = 0.0
      FVDIFF =  AMAX1(ABS(AMIN1( FV,               FVMAX ) )-FVMIN, FVMNDF)
      FYDIFF =  AMAX1(ABS(AMIN1(               FY, FYMAX ) )-FYMIN, FYMNDF)
      FFDDIF =  AMAX1(ABS(AMIN1( FV2/TOFFDZ+FY, FEGMAX) )-FEGMIN,FEGMND)
 206  IF( ATKALP -   161.0 ) 211, 207, 211
 207  ATKALP =  0.0
      GO TO 249

 211  IF( ATKALP ) 212, 210, 212
 212  IF(  ORDR  ) 213, 217, 213
 213  IF( ABS(ATTACK)  -  2.0   ) 215, 214, 214
 214  ATTACK =  SIGN( 3.5, ATKALP )
      IF( P125CH ) 2149,210,2149
2149  IERROR =  2149
      GO TO 999
 215  ATTACK =  SIGN(1.25+AMAX1(.0625-SIGN(.0625,ATKALP),P125CH),ATKALP)
      P125RG =  ( P125RG + SIGN(0.5,ATTACK) - 0.5 ) * AMIN1(P125RG,1.0)
      P125CH =  AMIN1( P125RG, 0.125 )
      CRSMAX =  ( SIGN(1.0,1.3125-ATTACK-P125CH) + 1.0 ) * CRNRG
      CRSMAX =  ZERON  + CRSMAX
 216  IF( RTEADJ ) 7151,2160,701
2160  ATKALP =  0.0
      GO TO 221
 217  ATKPHI =  0.0
      BANK   =  0.0
      IF( ABS(ATTACK)  -  2.0   ) 219, 2179,2179
2179  IERROR =  2179
      GO TO 999
 218  IF(  BNRG  ) 222, 219, 222
 219  ATKPHI =  0.0
      BANK   =  0.0
      GO TO 249
```



```
 221  IF( ATKPHI ) 222, 247, 222
 222  IF( ABS(ATKPHI)  - 12.0   ) 223, 231, 223
 223  IF(      ATKPHI  - 11.0   ) 2239,224, 2239
2239  IERROR =  2239
      GO TO 999
 224  IF( ABS(ATTACK)  -  1.125 ) 227, 227, 225
 225  IF( ABS(ATTACK)  -  1.375 ) 226, 226, 2259
2259  IERROR =  2259
      GO TO 999
 226  ATTACK = -SIGN( 3.5, ATTACK )
      GO TO 228
 227  ATTACK =  2.0
 228  ATKPHI =  - 0.0
      P125RG =  0.0
      P125CH =  0.0
      CRNRG  =  0.0
      CRSMAX =  0.0
      GO TO 247

 231  IF( ATKPHI ) 232, 2319,236
2319  IERROR =  2319
      GO TO 999
 232  IF(      ATTACK            ) 233, 244, 233
 233  IF(      ATTACK   -  2.0   ) 234, 244, 234
 234  IF( ABS(ATTACK)   -  1.25  ) 235,  45, 235
 235  IF( ABS(ATTACK)   -  3.5   ) 2359,245, 2359
2359  IERROR =  2359
      GO TO 999
 236  IF( ABS(ATTACK)   -  1.125 ) 241, 241, 237
 237  IF( ABS(ATTACK)   -  1.375 ) 243, 243, 2379
2379  IERROR =  2379
      GO TO 999
```



```
 241  ATAKNE =  0.0
 242  CRNRG  =  0.0
      CRSMAX =  0.0
      P125RG =  0.0
      P125CH =  0.0
      GO TO 246
 243  ATAKNE =  SIGN( 1.25, ATTACK )
      GO TO 242
 244  ATAKNE =  0.0
      IF( P125CH ) 2449,246, 2449
2449  IERROR =  2449
      GO TO 999
 245  ATAKNE =  SIGN( 1.25, ATTACK )
      IF( P125CH ) 2459,246, 2459
2459  IERROR =  2459
      GO TO 999
 246  ATTACK =  5.0
      ATZPHI =  0.0
      BRNG   =  0.0
      HIFTCP =  HIFT
      HIFTSP =  0.0
      SINPHI =  PHIU
      COSPHI =  1.0
      KONTOL =  KOUNT  -  1
      KOTCNG =  KOUNT
      PHICNG =  1.0
      GO TO 248
 247  ATAKNE =  ATTACK
 248  FTINIT =  FTI
      ALPSTR =  ALPHAD
      PHISTR =  PHIDEG
 249  RETURN
```



```
C     PITCH RATE DELAY AND ADJUSTMENT
 701  IF(      ATKSVE            ) 702, 713, 702
 702  IF( AND( ATKSVE, 0H100  ) ) 705, 703, 705
 703  NOALP  =  0
 704  IF( AND( ATKSVE, 0H40   ) ) 708, 713, 706
 705  NOALP  =  NOALP    +   1
 706  IF( AND( ATKSVE, 0H40   ) ) 708, 707, 708
 707  NOPHI  =  0
      GO TO 709
 708  NOPHI  =  NOPHI    +   1
 709  NO     =  NO       +   1
      NOTOT  =  NOTOT    +   1

 711  IF( NOALP -  NO ) 712, 721, 7199
 712  IF( NOPHI -  NO ) 713, 721, 7199
 713  NO     =  0
      NOALP  =  0
      NOPHI  =  0
 714  AVRAGE =  0.0
      TIMDIF =  0.0
7140  ALPDAT =  ALPONE
      PHIDAT =  PHIONE
 715  IF( GNMXAL ) 7151,7150,7151
7151  ALPOST =  ALPONE
      GAINAL =  GNMXAL
      ALPNEW =  ALPDAT-ALPOST*ATKALP
      IF( ALPNEW.LT.ALPSUM ) GAINAL = AMIN1(GNMXAL,GNMNAL)
      PTIAR3 =           -ALPOST*ATKALP   / GAINAL
      ALPHIO =           -ALPOST  *         ALPOST  /GAINAL*0.5*ATKALP
      PTIANE =  ALPNEW / GAINAL
      ALPNEU =  ( ALPDAT-ALPOST )*( ALPDAT+ALPOST )/GAINAL*0.5*ATKALP
      IF( GNMXPH ) 7152,7150,7152
7152  PHIOST =  PHIONE
      GAINPH =  GNMXPH
      PHINEW =  PHIDAT-PHIOST*ATKALP
      PHIKID =  ABS( PHINEW )
      IF( PHIKID.LT.PHISUM ) GAINPH = SIGN(AMIN1(GMAXPH,GMAXPHI,GNMXPH)
      PTIPHO =           -PHIOST*ATKALP   / GAINPH
      PHIHID =           -PHIOST  *         PHIOST  /GAINPH*0.5*ATKALP
      PTIPNE =  PHINEW / GAINPH
      PHINEW =  ( PHIDAT-PHIOST )*( PHIDAT+PHIOST )/GAINPH*0.5*ATKALP
7150  ATKALP =  0.0
      ATALIC =  OR( AND( ALPRAT/ALPDAT*DOORDR, 00377 ), PHIRAT )
      ATPHIC =  OR( AND( ALPRAT/PHIDAT*DOORDR, 00377 ), PHIRAT )
 716  IF( ABS(TCEXTS) .GT.  3.0 ) GO TO 718
      IF( JFAIL  ) 717, 221, 717
 717  IF( ABS(TCEXTS) .LT.  3.0 ) GO TO 221
 718  WRITE(6,701)  AVRAGE,TIMDIF,ALPDAT,PHIDAT,ATKSVE,NO,NOTOT
      IF( GNMXAL ) 7181,221, 7181
7181  WRITE(6,711)  ALPOST,PTIAR3,ALPHIO,PTIANE,ALPNEW,NOALP
      IF( GNMXPH ) 7182,221, 7182
7182  WRITE(6,721)  PHIDST,PTIPHO,PHIHID,PTIPNE,PHINEW,NOPHI
      GO TO 221
7199  IERROR =  7199
      GO TO 999
```

```
721  ALPSVE(NO)   =  ALPHAD
     PHISVE(NO)   =  PHIDEG
     TIMSVE(NO)   =  PTI
     IF( NO .LT. 91  ) GO TO 723
     DO 722  I = 71, NO
     ALPSVE(I-70) =  ALPSVE(I)
     PHISVE(I-70) =  PHISVE(I)
722  TIMSVE(I-70) =  TIMSVE(I)
     NO      =  NO      -  70
     NOALP   =  MAX0( NOALP - 70, 0 )
     NOPHI   =  MAX0( NOPHI - 70, 0 )
723  NI      =  NO      -  NDSUB
     IF( NI              ) 727, 727, 724
724  TIMDIF =  TIMSVE(NO)  -  TIMSVE(NI)
     IF( NOMAX-NOPHI ) 7241,725, 725
7241 NOPHI  =  NO
725  IF( NI -  NOALP ) 7199,731, 7251
7251 IF( NOMAX-NOALP ) 7252,726, 726
7252 NOALP  =  NO
     GO TO 731
726  IF( NO -  NOPHI ) 7199,741, 7199
727  IF( NO .LT. 2   ) GO TO 716
     NI      =  2       -  MOD( NO, 2 )
     TIMDIF =  TIMSVE(NI)  -  TIMSVE(NO)
     IF( TIMDIF ) 728, 716, 728
728  IF( NO -  NOALP ) 7199,7310,729
729  IF( NO -  NOPHI ) 7199,7410,7199

7310 AVRAGE =  ABS(    ( ALPSVE(NO)  -  ALPSVE(NI)  ) / TIMDIF    )
     GO TO 7140
731  ALPDAV =          ( ALPSVE(NO)  -  ALPSVE(NI)  ) / TIMDIF
     AVRAGE =  ABS( ALPDAV )
732  IF( TIMDIF.LT.TIMMIN ) ALPDAV = ( 2.0 - TIMDIF / TIMMIN ) * ALPDAV
     ALPDAV =  ALPDAV * AMAX0( MAXPRC - MULTNO * NO, MINPRC ) / 100.0
     ALPONT =  ALPDMN
     IF( AVRAGE.LT.ALPDM2 ) ALPONT = AMAX1( AVRAGE + ALPDM1, ALPONT )
     IF( GMXVAL.NE.0.0    ) ALPONT = AMAX1( ABS(ALPONE) / 4.0, ALPONT )
     ALPDAT =  AMIN1( AMAX1( ABS(ALPDAV), ALPONT ), ALPDMX )
     PHIDAT =  PHALRT * ALPDAT
     IF( ALPDAV*ATKALP .GT. 0.0  .AND.  ( ALPDAV .LT. 0.0
    1 .OR. ABS(RATMPY) .GT. 100.0 ) ) PHIDAT = PHIDAT * RATFAC
     IF( ABS( PHIDAT )  .GT. PHIDAX ) PHIDAT = PHIDMX
     IF( ABS( PHIDAT )  .LT. PHIDAN ) PHIDAT = PHIDMN
     IF( ALPDAT.GT.ALPDOS ) GO TO 715
     ITNORM =  ITNOOS(4)    +   ITNOOS(8)
     ITNORL =  ITNOOS(5)    +   ITNOOS(8)
     IF( ALPDAT.GT.ALPDM2 ) GO TO 715
     ITNORM =  ITNORM   +   ITNOOS(9)
     ITNORL =  ITNORL   +   ITNOOS(9)
     GO TO 715
```



```
7410 AVRAGE =  - ABS(    ( PHISVE(NO)  -  PHISVE(NI)  ) / TIMDIF    )
     GO TO 7140
741  PHIDAV =           ( PHISVE(NO)  -  PHISVE(NI)  ) / TIMDIF
     AVRAGE =  - ABS( PHIDAV )
742  IF( TIMDIF.LT.TIMMIN ) PHIDAV = ( 2.0 - TIMDIF / TIMMIN ) * PHIDAV
     PHIDAV =  PHIDAV * AMAX0( MAXPRC-MULTNO*NO,MINPRC ) / 100.0
     PHIDNT =  PHIDAN
     IF( AVRAGE.GT.PHIDA2 ) PHIDNT = AMAX1( - PHIDA1 - AVRAGE,PHIDNT )
     IF( BANKPH.NE.0.0    ) PHIDNT = AMAX1( ABS(PHIONE) / 4.0, PHIDNT )
     PHIDAT =  SIGN( AMIN1(AMAX1(ABS(PHIDAV),PHIDNT),PHIDAX ), PHIDMX )
     ALPDAT =  ALPHRT * PHIDAT
     IF( PHIDAV*ATKALP*PHIULT .LT. 0.0  .AND.  ( PHIDAV*PHIULT .GT. 0.0
    1 .OR. ABS(RATMPY) .GT. 100.0 ) ) ALPDAT = ALPDAT / RATFAC
     ALPDAT =  AMIN1( AMAX1( ALPDAT, ALPDMN ), ALPDMX )
     IF( ALPDAT.GT.ALPDOS ) GO TO 715
     ITNORM =  ITNOOS(4)    +   ITNOOS(8)
     ITNORL =  ITNOOS(5)    +   ITNOOS(8)
     IF( ALPDAT.GT.ALPDM2 ) GO TO 715
     ITNORM =  ITNORM   +   ITNOOS(9)
     ITNORL =  ITNORL   +   ITNOOS(9)
     GO TO 715
```

```
C     ERROR MESSAGE FOR INCORRECT CONDITIONS WITHIN PROGRAM
 999  WRITE(6,99)  (ERROR,N,IBPSTD,COUNT,KONTRL,NUMCAS,ATTACK,ATANNE,
    1   CASRAT,ATKALP,ATKPHI,PTI,PTINIT,ALPHAD,ALPOLD,PHIDEG,PHIOLD,
    2   ATTACK,ATANNE,CASKST,ATKALP,ATKPHI
      CALL OUTACK
      CALL EXIT
      STOP


      END
```

```
$IDPTC DCLN    PULIST,PMAP,DD,DECK
      SUBROUTINE PITCH

C     SUBROUTINE PITCH  CODE NUMBER PCTCH                  VERSION 6.0

C     THIS PROGRAM WILL ADJUST THE ATTITUDE OF THE VEHICLE SO AS TO
C     MINIMIZE TEMPERATURE AND ACCELERATION PEAKS AND MAXIMIZE CROSS
C     RANGE.

C     COMMON STATEMENTS FOR GLOBAL VARIABLES
      COMMON /CPVAR/   PV2,PDUMV2,PV,PDUMUR(3),PY,PDUMM(11),PTI,PDUMD(7)
      COMMON /CSCAP/   ALPHAD,ALPOLD,SINALP,COSALP,PHIDEG,PHIOLD,SINPHI,
     1                 COSPHI
      COMMON /CINTS/   N,M,NTP,IOPDVD,IDUMMG,KOUNT,INTLST,ILAST,JFAIL,
     1                 IDUMM(11)
      COMMON /CPIDS/   PDUMP(2),PIDG2,PDUMM(7),RDUMMA,CGRMR,PDUMB(6)

C     COMMON STATEMENTS FOR SUBROUTINE VARIABLES
      COMMON /CPTTDS/  PTDUMP(4),DENSTY,PTDUMP(8)
      COMMON /CRANGE/  RDUMP(4),PSIDSA
      COMMON /CGRAV/   GRDUMP(19),GREPR2,TOAPR2
      COMMON /CPITCH/  ATTACK,ATAKNE,ATKALP,ATKPHI,BANK,CRSMAX,ROLL,
     1                 COSMN,COSD2M,CLSQRM,WOD,DRAG,HIPT,HIPTSP,HIPTCP,
     2                 PSIPST,PSIPOC,CSCPOR,ALPDAT,PHIDAT,PHIRTE,BIASPR

C     COMMON STATEMENTS FOR LOCAL VARIABLES
      COMMON /CPITVR/  PTINIT,KONTOL,KOTCNS,PHICNG,WOS,CDEIND,CP,CPDRSG,
     1   ALPHAU,COSOMU,COS2MU,CLS2MU,ALPHAL,COSOML,COS2ML,CLS2ML,PHIF,
     2   ALPHAF,COSOMF,COS2MF,CLS2MF,PHIU,PHIULT,PHIL,SINPHL,COSPHL,
     3   PHILET,PHILMN,PHILOP,PVTMAX,PVTMAZ,PEGTME,ICRSME,ICRSVE,
     4   PVMIN,PVPIN,PEGMIN,PVDIPF,PVDIFF,PDDDIF,PTZSCAL,SINPMF,COSPMF,
     5   GAMVAL,GAINAL,ALPSTR,ALPNEW,ALPDAT,ALPONE,ALPOST,PTIANE,
     6   GAMVPH,GAINPH,PHISTR,PHINEW,PHIDAT,PHIONE,PHIOST,PTIPNE,PHIOTR

C     LOCAL DATA VALUES WHICH ARE FIXED
      DATA   DRAD,DMIC/0/0400000000,0100000000/
      DATA   O37627,O4/317/O777477000000,O77773700000/
```



```
      IF( ATKALP + ATKPHI )   1, 291,   1
    1 IERROR =  1
      IF( KOUNT  +  1 ) 999, 250, 999
  250 RETURN

  251 IF( ATTACK  - ATAKNE ) 252, 261, 252
  252 IF(    N    -    2   ) 2529,253, 2529
 2529 IERROR =  2529
      GO TO 999
  253 IF(   IOPDVD          ) 2539,254, 2539
 2539 IERROR =  2539
      GO TO 999
  254 IF( KONTOL+1 - KOUNT ) 2549,257, 255
 2549 IERROR =  2549
      GO TO 999
  255 IF( KONTOL  -  KOUNT ) 2559,256, 2559
 2559 IERROR =  2559
      GO TO 999
  256 ATAKNE =  ATTACK
      IF( PHICNG ) 2569,259, 2569
 2569 IERROR =  2569
      GO TO 999
  257 ATTACK =  ATAKNE
      IF( PHICNG ) 258, 259, 258
  258 PHICNG =  0.0
      PHIDEG =  PHIU
      PHIOLD =  PHIU
      IF( KOTCNS  -  KOUNT ) 2589,259, 2589
 2589 IERROR =  2589
      GO TO 999
  259 KONTOL =  - 1
      KOTCNS =  - 1
      IF( ATAKNE ) 262, 264, 262

  261 IF( ATAKNE ) 262, 264, 262
  262 PTIDIF =  PTI  -  PTINIT
  263 IF( ABS(ATAKNE)  -  2.9   ) 301, 301, 301
  264 PTINIT =  2.9E75
  265 IF( CRSMAX ) 267, 290, 263
  266 IF( CRSMAX ) 267, 2669,267
 2669 IERROR =  2669
      GO TO 999
  267 PHIDEG =  PHILET
      GO TO 404
```



```
  301 IF( AND( ATAKNE, DMICO  ) ) 302, 341, 302
  302 IF( ATAKNE ) 321, 3029,331
 3029 IERROR =  3029
      GO TO 999

  321 IF( GAMVAL ) 3211,3219,3211
 3211 ALPONE =  ALPOST  -  GAINAL * PTIDIF
 3212 IF(-ALPONE.GT.ALPDAT ) GO TO 3215
 3214 ALPHAD =  ALPSTR  +  ALPOST * PTIDIF  -  0.5 * GAINAL * PTIDIF**2
      GO TO 3216
 3215 ALPHAD =  ALPSTR  +  ALPNEW  -  ALPDAT * ( PTIDIF - PTIANE )
      ALPONE =  - ALPDAT
 3216 IF( ALPHAD -  ALPHAL ) 3217,3218,351
 3217 ALPHAD =  ALPHAL
 3218 ALPONE =  0.0
      GO TO 324
 3219 ALPHAD =  ALPSTR  -  ALPDAT * PTIDIF
  322 IF( ALPHAD  - ALPHAL ) 323, 324, 351
  323 ALPHAD =  ALPHAL
  324 IF( ALPHAD  - ALPOLD ) 329, 341, 329
  329 ALPOLD =  ALPHAL
      COSD2M =  COS2ML
      CLSQ2M =  CLS2ML
      GO TO 341

  331 IF( GAMVAL ) 3311,3319,3311
 3311 ALPONE =  ALPOST  +  GAINAL * PTIDIF
 3312 IF( ALPONE.GT.ALPDAT ) GO TO 3315
 3314 ALPHAD =  ALPSTR  +  ALPOST * PTIDIF  +  0.5 * GAINAL * PTIDIF**2
      GO TO 3316
 3315 ALPHAD =  ALPSTR  +  ALPNEW  +  ALPDAT * ( PTIDIF - PTIANE )
      ALPONE =  ALPDAT
 3316 IF( ALPHAU -  ALPHAD ) 3317,3318,351
 3317 ALPHAD =  ALPHAU
 3318 ALPONE =  0.0
      GO TO 334
 3319 ALPHAD =  ALPSTR  +  ALPDAT * PTIDIF
  332 IF( ALPHAU  - ALPHAD ) 333, 334, 351
  333 ALPHAD =  ALPHAU
  334 IF( ALPHAD  - ALPOLD ) 335, 341, 335
  335 ALPOLD =  ALPHAU
      COSD2M =  COS2MU
      CLSQ2M =  CLS2MU
```



```
  341 IF(    N    -     1   ) 3419,342, 393
 3419 IERROR =  3419
      GO TO 999
  342 IF( IOPDVD +  200    ) 3429,343, 3429
 3429 IERROR =  3429
      GO TO 999
  343 KONTOL =  KOUNT
  344 ATAKNE =  AND( ATAKNE, O37627 )
  345 IF( ABS(ATAKNE)  -  1.0    ) 346, 348, 346
  346 IF( ABS(ATAKNE)  -  1.125 ) 347, 402, 347
  347 IF( ABS(ATAKNE)  -  3.0   ) 3479,349, 3479
 3479 IERROR =  3479
      GO TO 999
  348 ATAKNE =  - 0.0
      IF( CRSMAX ) 267, 401, 267
  349 ATAKNE =  2.0
      GO TO 391

  391 CALL SNCSALF 03911 (
      ALPHAR =  ALPHAD * DEGRAD
      SINALP =  SIN(ALPHAR)
      COSALP =  COS(ALPHAR)
 3911 CONTINUE
  392 CONTINUE
      CL      =  ABS( SINALP ) * SINALP
      CD      =  CDZERO   +   CPDRAG * CL * SINALP
      CL      =  CP * CL * COSALP
      WOD     =  CL / CD
      CDSQM   =  CD * GREPR2 / WOS
      COSD2M  =  CDSQM / 2.0
      CLSQRM  =  COSD2M * WOD
  393 CONTINUE
      IF( ABS(ATAKNE)  -  2.9   ) 401, 3939,901
 3939 IERROR =  3939
      GO TO 999

  361 IF( ABS(ATAKNE)  -  1.125 ) 3619,402, 3619
 3619 IERROR =  3619
      GO TO 999
```

```
401  IF( AND( ATAKNE, OM40   ) ) 402, 471, 402
402  IF( ATAKNE ) 403, 4029,431
4029 IERROR =  4029
     GO TO 999
403  IF( CRSMAX ) 404, 421, 404
436  GO TO(411,412,413,414,415,416,417,418,419,4049,405,406,4071),ICRSMX
4049 IERROR =  4049
     GO TO 999
405  IF( FV   .GT.  FVTMAX ) GO TO 420
     GO TO 408
406  IF( FY   .GT.  FYTMAX ) GO TO 420
     GO TO 408
407  IF( FV2 / TGRFAZ + FY   .GT.  FEGTHC ) GO TO 420
408  ICRSMX =  ICRSVF
     ICRSVF =  19
     ALPHAL =  ALPHAF
     COS2HL =  COSO2F
     COS2HL =  COS2HF
     CLS2HL =  CLS2HF
     PHIL   =  PHIF
     PHILET =  PHIF
     SINPHL =  SINPHF
     COSPHL =  COSPHF
     GO TO 420

411  PHIL   =  PHILMN+PHILOF*SQRT(AMIN1(AMAX1(FV-FVMIN,0.)/FVDIFF,1.0))
     GO TO 420
412  PHIL   =  PHILMN  +  PHILOF * AMIN1(AMAX1(FV-FVMIN,0.)/FVDIFF,1.0)
     GO TO 420
413  FVMULT =                          AMIN1(AMAX1(FV-FVMIN,0.)/FVDIFF,1.0)
     PHIL   =  PHILMN  +  PHILOF * FVMULT * SQRT(FVMULT)
     GO TO 420
414  PHIL   =  PHILMN+PHILOF*AMIN1(AMAX1(FV-FVMIN,0.)/FVDIFF,1.0) ** 2
     GO TO 420
415  PHIL   =  PHILMN+PHILOF*SQRT(AMIN1(AMAX1(FY-FYMIN,0.)/FYDIFF,1.0))
     GO TO 420
416  PHIL   =  PHILMN  +  PHILOF * AMIN1(AMAX1(FY-FYMIN,0.)/FYDIFF,1.0)
     GO TO 420
417  FYMULT =                          AMIN1(AMAX1(FY-FYMIN,0.)/FYDIFF,1.0)
     PHIL   =  PHILMN  +  PHILOF * FYMULT * SQRT(FYMULT)
     GO TO 420
418  PHIL   =  PHILMN+PHILOF*AMIN1(AMAX1(FY-FYMIN,0.)/FYDIFF,1.0) ** 2
     GO TO 420
419  PHIL   =  PHILMN  +  PHILOF * AMIN1(AMAX1(FV2/TGRFAZ+FY-FEGMIN,
    1                                              0.)/FEGDIF,1.0)
420  IF( AND( ATAKNE, OM40   ) ) 421, 422, 421
```



```
421  IF( GAMNPH ) 4211,4219,4211
4211 PHIDNE =  PHIDST  -  GAINPH * FTIDIF
     IF( PHILET ) 4212,4029,4213
4212 IF(-PHIDNE.GT.PHIDAT ) GO TO 4215
     GO TO 4214
4213 IF(-PHIDNE.LT.PHIDAT ) GO TO 4215
4214 PHIDEG =  PHISTR  +  PHIDST * FTIDIF  -  0.5 * GAINPH * FTIDIF**2
     GO TO 4216
4215 PHIDEG =  PHISTR  +  PHINEW  -  PHIDAT * ( FTIDIF - FTIPNE )
     PHIDNE =  - PHIDAT
4216 IF( ( PHIL - PHIDEG ) [illegible] PHIULT ) 4217,4218,451
4217 PHIDEG =  PHIL
4218 PHIDNE =  0.0
     GO TO 424
4219 PHIDEG =  PHISTR  -  [illegible]DAT * FTIDIF
422  IF( ( PHIL - PHIDEG ) * PHIULT ) 423, 424, 451
423  PHIDEG =  PHIL
424  IF( PHIDEG  - PHIOLD ) [illegible], 427, 425
425  IF( CRSMAX ) 426, 428, [illegible]
426  CALL INCSPH( 84261 )
     PHIRAD =  PHIDEG * DGDRC*
     SINPHI =  SIN( PHIRAD )
     COSPHI =  COS( PHIRAD )
4261 CONTINUE
427  IF( AND( ATAKNE, OM40   ) ) 441, 441, 441
428  IF( PHIDEG -  PHILET ) 4289,429, 4289
4289 IERROR =  4289
     GO TO 999
429  PHIOLD =  PHILET
     SINPHI =  SI[illegible]
     COSPHI =  CO[illegible]
     GO TO 441

431  IF( GAMNPH ) 4311,4319,4311
4311 PHIDNE =  PHIDST  +  GAINPH * FTIDIF
     IF( PHILET ) 4312,4029,4313
4312 IF( PHIDNE.GT.PHIDAT ) GO TO 4315
     GO TO 4314
4313 IF( PHIDNE.LT.PHIDAT ) GO TO 4315
4314 PHIDEG =  PHISTR  +  PHIDST * FTIDIF  +  0.5 * GAINPH * FTIDIF**2
     GO TO 4316
4315 PHIDEG =  PHISTR  +  PHINEW  +  PHIDAT * ( FTIDIF - FTIPNE )
     PHIDNE =  PHIDAT
4316 IF(           PHIDEG   * PHIULT ) 4317,4317,451
4317 PHIDEG =  PHIU
4318 PHIDNE =  0.0
     GO TO 434
4319 PHIDEG =  PHISTR  +  PHIDAT * FTIDIF
432  IF(           PHIDEG   * PHIULT ) 433, 433, 451
433  PHIDEG =  PHIU
434  IF( PHIDEG  - PHIOLD ) 435, 441, 435
435  PHIOLD =  PHIU
     SINPHI =  PHIU
     COSPHI =  1.0
```



```
441  IF(    N    -    1   ) 4419,442, 493
4419 IERROR =  4419
     GO TO 999
442  IF( IGPDVD -  200    ) 4429,443, 4429
4429 IERROR =  4429
     GO TO 999
443  KONTRL =  KOUNT
444  ATAKNE =  AND( ATAKNE, 047717 )
     IF( ABS(ATAKNE)  -  1.0   ) 445, 448, 445
445  IF( ABS(ATAKNE)  -  1.25  ) 446, 461, 446
446  IF( ABS(ATAKNE)  -  2.0   ) 447, 448, 447
447  IF( ABS(ATAKNE)  -  3.5   ) 4479,449, 4479
4479 IERROR =  4479
     GO TO 999
448  ATAKNE =  - 0.0
     GO TO 461
449  ATAKNE =  SIGN( 1.25, ATAKNE )
     GO TO 461

451  CALL INCSPH( 84511 )
     PHIRAD =  PHIDEG * DGDRDR
     SINPHI =  SIN( PHIRAD )
     COSPHI =  COS( PHIRAD )
4511 CONTINUE
452  CONTINUE
493  CONTINUE

461  RETURN

471  IF( CRSMAX ) 472, 461, 472
472  IF( PI29CH ) 404, 267, 404
```



```
901  IF(   DOBLE( BANK )   ) 921, 931, 921

921  COSPNE =  1.0  +  (PSIGMA - PIGAZ) * PHIRTE / CLSGRN / DENSTY / FV
922  IF( COSPHI -  COSPNE ) 923, 991, 993
923  IF( 1.0    -  COSPNE ) 924, 924, 991
924  PHIDEG =  PHIU
     PHIOLD =  PHIU
     SINPHI =  PHIU
     COSPHI =  1.0
     GO TO 953

931  PHIDEG =  PHISTD  +  PHIDTE * FTIDIF
     GO TO 492

991  COSPHI =  COSPNE
992  SINPHI =  SIGN( SQRT( AMAX( AMAX(1.0-COSPHI)*(1.0+COSPHI) ) ), PHIU)
     PHIDEG =  1000.0
     PHIOLD =  2000.0
     IF( COSPHI ) 9929,9929,953
9929 IERROR =  9929
     GO TO 999
953  CONTINUE

961  RETURN




C       FORMAT
   99   FORMAT(  1H0       IERROR =, 6I15  /  11F11.4  /  8O15  )

C       ERROR MESSAGE FOR INCORRECT CONDITIONS WITHIN PROGRAM
  999   WRITE(6,99)  IERROR,N,IGPDVD,KOUNT,KONTRL,NOTCHS,ATTACK,ATAKNE,
      1    CRSMAX,ATKALP,ATKPHI,FTI,FTINIT,ALPHAO,ALPOLD,PHIDEG,PHIOLD,
      2    ATTACK,ATAKNE,CRSMAX,ATKALP,ATKPHI
        CALL ENTKCK
        CALL EXIT
        STOP


        END
```

31


```
$IBMAP MCPALL  100,00,DECK
*                                                         VERSION 6.0
        LBL     MCPALL,BEGIN,
        DETAIL  /,/0
*       THE FOLLOWING SUBROUTINES ARE THE ENTRY POINTS FOR THE MINIMUM
*       POSSIBLE CONFIGURATION OF THESE SUBROUTINES.
        ENTRY   SNCSOA
        ENTRY   SNCSOE
        ENTRY   SNCSPS
        ENTRY   SNCSPN
        ENTRY   SNCSAL
        ENTRY   SWITC
SNCSOA  TRA     1,4
SNCSOE  TRA     1,4
SNCSPS  TRA     1,4
SNCSPN  TRA     1,4
SNCSAL  TRA     1,4
SWITCH  ZAC
        TRA     1,4
        SPACE   30
```



```
*       THE FOLLOWING SUBROUTINES CONTAIN VITAL PROGRAMMING FEATURES
*       WHICH MUST BE PRESENT IF USED.   THESE DUMMY ENTRY POINTS WILL
*       HALT COMPUTATION IF THEY ARE USED INSTEAD OF THE ACTUAL ROUTINE
*       WITH THE PROPER PROGRAMMING FEATURES.
        ENTRY   CHECKR
        ENTRY   CHECKO
        ENTRY   CHECK1
        ENTRY   CHECK2
        ENTRY   CHECKT
        ENTRY   GRAVCS
        ENTRY   PTTOS
        ENTRY   PTTOSI
        ENTRY   RADIUS
        ENTRY   RANGE
        ENTRY   PITCH
        ENTRY   PITCHI
        ENTRY   PITCHA
CHECKR  CLA     WRDCKR
        TRA     MESAGE+2
WRDCKR  BCI     1,CHECKR
CHECKO  CLA     WRDCKO
        TRA     MESAGE+2
WRDCKO  BCI     1,CHECKO
CHECK1  CLA     WRDCK1
        TRA     MESAGE+2
WRDCK1  BCI     1,CHECK1
CHECK2  CLA     WRDCK2
        TRA     MESAGE+2
WRDCK2  BCI     1,CHECK2
CHECKT  CLA     WRDCKT
        TRA     MESAGE+2
WRDCKT  BCI     1,CHECKT
GRAVCS  CLA     WRDGRV
        TRA     MESAGE+2
WRDGRV  BCI     1,GRAVCS
PTTOS   CLA     WRDPTO
        TRA     MESAGE+2
WRDPTO  BCI     1,PTTOS
PTTOSI  CLA     WRDPTI
        TRA     MESAGE+2
WRDPTI  BCI     1,PTTOSI
RADIUS  CLA     WRDRAD
        TRA     MESAGE+2
WRDRAD  BCI     1,RADIUS
RANGE   CLA     WRDRNG
        TRA     MESAGE+2
WRDRNG  BCI     1,RANGE
PITCH   CLA     WRDPCH
        TRA     MESAGE+2
WRDPCH  BCI     1,PITCH
PITCHI  CLA     WRDPCI
        TRA     MESAGE+2
WRDPCI  BCI     1,PITCHI
PITCHA  CLA     WRDPCA
        TRA     MESAGE+2
WRDPCA  BCI     1,PITCHA
```



```
*       MESSAGE GENERATING SECTION FOR IMPROPERLY CALLED SUBROUTINES
        ENTRY   MESAGE
MESAGE  CLA*    3,4
        STL     RETURN
        STO     FORMAT+3
        SXA     SYSLOC,4
        STO     SYSLOC,0
        SXA     LK.DR,4
        SXA     SAVE4,4
        TSX     .FWRD.,4
        TXI     *+4,,2
        PZE     11,,LK.DR
        PZE     .UN06.
        PZE     FORMAT
        TSX     .FFIL.,4
        ZAC
        DCT     RETURN
        TSX     .EXIT.,4
        TXI     *+2,,0
        PZE     11,,LK.DR
        STZ     RETURN
SAVE4   AXT     **,4
        STO     SYSLOC,4
        TRA     1,4
        SPACE   1
RETURN  PZE     0
FORMAT  BCI     9,( 1H0/3H0ERROR - XXXXXX SUBROUTINE NOT PRESENT.)
        SPACE   2
LK.DR   PZE     0
        BCI     1,MCPALL
        SPACE   17
```



```
*       THE FOLLOWING SUBROUTINES ARE THE DUMMY ENTRY POINTS WHICH ALLOW
*       CONTROL TO PROCEED THROUGHT THEM WHEN THEY ARE USED IF NO ACTUAL
*       VITAL PROGRAMMING FEATURES ARE NECESSARY.
        ENTRY   CONSTS
        ENTRY   DENVAR
        ENTRY   DENVRI
        ENTRY   GRAVCI
        ENTRY   GRAVCR
        ENTRY   GRAVCP
        ENTRY   PAUSE
        ENTRY   RANGEI
        ENTRY   RANGER
        ENTRY   FINAL
        ENTRY   PRGLST
        ENTRY   RGMLST
        ENTRY   TPPLST
        ENTRY   ENMLST
CONSTS  TRA     1,4
DENVAR  SXA     SYSLOC,4
        STO     SYSLOC,0
        SXA     LK.DR,4
        SXA     *+4
        TSX     PTTOS,4
        TXI     *+2,,0
        PZE     1,,LK.DR
        AXT     **,4
        STO     SYSLOC,4
        TRA     1,4
DENVRI  SXA     SYSLOC,4
        STO     SYSLOC,0
        SXA     LK.DR,4
        SXA     *+4
        TSX     PTTOSI,4
        TXI     *+2,,0
        PZE     2,,LK.DR
        AXT     **,4
        STO     SYSLOC,4
        TRA     1,4
GRAVCI  TRA     1,4
GRAVCR  TRA     1,4
GRAVCP  TRA     1,4
PAUSE   TRA     1,4
RANGEI  TRA     1,4
RANGRC  TRA     1,4
FINAL   TRA     1,4
PRGLST  TRA     1,4
RGMLST  TRA     1,4
TPPLST  TRA     1,4
ENMLST  TRA     1,4
        SPACE   6
        END
```

```
$IBFTC PCLS     FULIST,PMAP,DD,DECK
      SUBROUTINE PRGLST

C     SUBROUTINE PRGLST CODE NUMBER PCGLST                     VERSION 6.0

C     THIS PROGRAM IS DESIGNED TO LIST THE BINARY DECKS BEING USED ON
C     THE OUTPUT TAPE AND TO ALSO LIST ON THE OUTPUT TAPE AS WELL AS
C     WRITING ON SCRATCH TAPE NUMBER TWO (2) THE DATA TO BE USED DURING
C     THE RUN.   SCRATCH TAPE NUMBER TWO (2) IS REWOUND BEFORE AND AFTER
C     THE PROGRAM DATA IS WRITTEN ON THE TAPE.   A NORMAL DATA CARD,
C     BUT WITHOUT THE DOLLAR ($) SIGN IN COLUMN ONE (1) MUST PRECEED
C     THE PROGRAM DATA AND  AN **END OF DATA** CARD MUST FOLLOW THE
C     PROGRAM DATA.

      REAL   CARDS(15),IDLDR
      DATA   CHLWRD,DATA,DOLLAR,END/2H$DATA,1H$,5HEND OF/
      DATA   SKIP,IDLDR,THEFOL/1H1,5H$IBLDR,5HTHE FO/
    1 FORMAT(        A1, A5, 12A6, A2  )
    2 FORMAT(  1X, A1, A5, 12A6, A2  )

      REWIND 2
   11 READ (1,1)   CARDS
      IF( CHLWRD .EQ. END       ) WRITE(2,1) CARDS
      IF( CHLWRD .NE. DATA     ) GO TO 12
      IF( IDLDR  .EQ. CARDS(2)  ) CARDS = DOLLAR
      IF( DATA   .EQ. CARDS(2)  ) CARDS = DOLLAR
      IF( THEFOL .EQ. CARDS(2)  ) WRITE(6,1) SKIP
   12 WRITE(6,2)   CARDS
      IF( CHLWRD .NE. CARDS(2)  ) GO TO 11
      IF( CHLWRD .EQ. END       ) GO TO 13
      CHLWRD  =  END
      GO TO 11
   13 ENDFILE 2
      REWIND 2
      RETURN




      END
```



```
$DATA

THE FOLLOWING DATA CARDS WILL BE USED DURING THIS RUN

 DATA
     0.009       0.02  5.0     0     0    -1     0    -1   -0    -1    -1    -1  90001
   0    -0     1    -1     1     0    -0     1     1     3                         90001
9000110-1-0 1 0 0-1 6 -16777 3-1090.6          -10.0           -0.0TEMP LOGIC 1
     37.17       40.0     0.0625       1.75     1.0075        1.0999999999.  BCP- 27A
.069013171.139626341.209439511          0.0        1.0     1000.0         0.0  BCP- 27B
     45.0          0.6.785398164.392699082 1.57079633         0.0         1.5  BCP- 27C
   1000.0    13000.0      500.0   100000.0   200000.0    10000.0        10.0  BCP- 27D
 1000000.0 900000.00  1000000.0     0.0625       0.75      0.075         1.5  BCP- 27E
      1.0        4.0          4         10         10        110         310  BCP- 27F
     0.75        1.5        1.0        2.0    25000.0   250000.0   900000.0  BCP- 27P
     45.0        0.0       20.0        0.0      -75.0       95.0         0.0 0BCP- 6TJ
     5000       2000          0       1000     3  0.3   0.5  0.5               BCP-  1J
        0    1000000          0      10000     250000        1.0           1  BCP-  2J
       -1    1020000    1000000     100000     200000          0       50000  BCP-  6J
  -1    95     0     0    -5    10    1.9001     3     5                       BCP- 10A
     -200       -400        -10       -100        -10       -200              BCP- 10B
  16    96 2100    15  1000    10   6  0.25   10     4                         BCP- 10L
    86400      86400        600        600      36000         20          50  BCP- 20A
    2.000      2.000      1.000      1.000     324700                         BCP- 20B
  10     0    30     9    30    -1    25  1000       .0.000000001              BCP- 20L
       -1        360        180       -100       -360         45          -1  BCP- 40A
       -1        720        360       -360       -720          5              BCP- 40K
      2.0        1.5        1.5      1.975        1.5        1.0              BCP-100A
      0.5        0.5        0.5        0.5        2.0     1000.0              BCP-100B
      0.5        0.5    0.70125      0.625    0.70125        0.0              BCP-100C
 1000.000   1000.000   1000.000   1000.000   1000.000      1.000              BCP-100D
        0       -100       -1.0      -1500       -100      -0.25       -1500  BCP-100E
     -200       -100        -20       -360       -200       -500              BCP-100F
        0      10000          0                                               BCP-100P
      0.014391000                  1.00070918100.726179942 00                 BCP-300A
         32.1700                    32.17              239.91                 BCP-300B
      0.001140394        26013.990063143       20000.000000000    250000.0  BCP-300L
  1.5  0.1 -0.5  0.3 1050000.0        0.0        0.1    50000.0   250000.0  BCP-400J
      26000.000      350000.000      1.000      35.300      0.000   -230.000  90001
9000211-1-0 1 0 0-1 6999  -0 3-1090.0          -10.0           -0.0TEMP LOGIC 2

END OF DATA
```

PROGRAM D

Sample Output

CASE NUMBER 3312 LETTER CODE FOTPRNT10=45 1. -1. 1. 1. 0. 0. -1. 3. 999.004000 6

ANGINC = 0.005000 ATTINC = 0.020000 TEPINC = 10.00 0. 0.-1. 0.-1.-0. 1.-1.-1. 0.-0. 1.-1. 1. 0.-0. 1. 1. 2.

PHI = -50.000 DEGREES CD S/M = 0.1283230 SQ.FT./SLUG L/D = 1.5075630 ANGLE OF ATTACK = 22.70615 DEGREES
COEFFICIENTS - LIFT = 0.2405408 DRAG = 0.1595561

NOTE - IN ORDER TO MINIMIZE THE TEMPERATURE AND ACCELERATION PEAKS, THE ATTITUDE OF THE VEHICLE WILL BE SLOWLY CHANGED TO THE FOLLOWING VALUES IN PLACE OF THE NOMINAL (OR TERMINAL) VALUES GIVEN ABOVE WHENEVER THE EQUILIBRIUM STAGNATION POINT TEMPERATURE IS INCREASING. THE ATTITUDE WILL SLOWLY CHANGE BACK TO THE NOMINAL VALUES WHENVER THE TEMPERATURE IS DECREASING. THE FOLLOWING RATES OF CHANGE WILL BE USED -

MAXIMUM PITCH RATE =	4.000 DEGREES/SEC.	MAXIMUM ROLL RATE =	8.000 DEGREES/SEC.
MINIMUM PITCH RATE =	0.075 DEGREES/SEC.	MINIMUM ROLL RATE =	0.150 DEGREES/SEC.
UPPER PITCH ACCEL. =	10.000 DEGREES/SEC**2	UPPER ROLL ACCEL. =	30.000 DEGREES/SEC**2
LOWER PITCH ACCEL. =	1.000 DEGREES/SEC**2	LOWER ROLL ACCEL. =	2.000 DEGREES/SEC**2

PHI = -0.000 DEGREES CD S/M = 0.7962477 SQ.FT./SLUG L/D = 0.6803012 ANGLE OF ATTACK = 55.00000 DEGREES
COEFFICIENTS - LIFT = 0.6735322 DRAG = 0.9900500

NOTE - IN ORDER TO MAXIMIZE CROSS RANGE, THE NOMINAL ANGLE OF ROLL (NOMINAL MAXIMUM IF TEMPERATURES ARE BEING MINIMIZED) WILL VARY (ACCORDING TO THE 1.0 POWER OF THE VELOCITY) BETWEEN -50.000 DEGREES AND -0.000 DEGREES OVER A SPECIFIED VELOCITY RANGE.

NOTE - IN ORDER TO EXTEND (OR SHORTEN) THE DOWN RANGE, THE FOLLOWING PAIRS OF LIMITS WILL BE USED FOR THE ANGLES OF ATTACK AND ROLL UNTIL THE EQUILIBRIUM STAGNATION POINT TEMPERATURE STARTS ITS SECOND RISE (OR SOME OTHER APPROPIATE CONDITION IS REACHED IN THE FLIGHT) -

PHI = -75.000 DEGREES CD S/M = 0.1045544 SQ.FT./SLUG L/D = 1.4795666 ANGLE OF ATTACK = 20.00000 DEGREES
COEFFICIENTS - LIFT = 0.1923655 DRAG = 0.1300148

PHI = -0.000 DEGREES CD S/M = 0.5300983 SQ.FT./SLUG L/D = 0.9387019 ANGLE OF ATTACK = 45.00000 DEGREES
COEFFICIENTS - LIFT = 0.6187184 DRAG = 0.6591213

INITIAL VELOCITY = 26000.000000000000 00 FT/SEC. INITIAL GEOCENTRIC RADIUS = 21275839.895013120D 00 FEET

VELOCITY (FT/SEC)	ALTITUDE (FEET)	GLIDE ANGLE (DEG)	AZIMUTH (DEG)	LATITUDE (DEG)	LONGITUDE (DEG)	TIME (SEC)	ANGLES OF ATTACK (DEG)	ROLL (DEG)	NO.	ACCEL (G'S)	TEMP(S) (DEG-R)	CNT	DOWNR (DEG)	CROSSR (DEG)	PSIRG (DEG)
26000.	350000.	1.000000	55.0000	0.	-220.0000	0.	45.	-0.	1	0.002	1915.412	792	0.	0.	0.
26023.181	300000.	0.735110	55.3146	5.0064	-212.8130	124.750	45.	-0.	2	0.035	2682.115	705	8.7513	0.	0.
25769.898	275111.57	0.031124	56.2930	10.0382	-205.3566	252.382	45.	-0.	3	0.143	3177.035	748	17.6914	0.	0.
25756.014	275069.88	-0.	56.3781	10.4143	-204.7828	262.129	20.	-75.	4	0.037	3176.616	718	18.3699	0.00009	-0.0168
25589.385	277714.71	0.	60.2090	20.2233	-188.4561	529.230	20.	-75.	5	0.031	3102.301	718	36.8983	0.09142	-0.5276
25582.902	277700.32	0.005658	60.4460	20.0179	-187.7172	540.757	20.	-75.	6	0.031	3101.995	749	37.6953	0.09889	-0.5471
25539.435	277508.73	0.008254	61.7324	22.5969	-183.8645	600.	24.3876	-62.7763	7	0.044	3102.011	728	41.7871	0.14234	-0.6715
25521.288	277429.91	0.008106	62.2547	23.3205	-182.3870	622.336	24.2598	-63.0318	8	0.044	3101.991	742	43.3279	0.16107	-0.7215
25455.791	277133.46	0.009467	64.1365	25.6196	-177.3904	695.313	26.1342	-59.2831	9	0.051	3102.166	742	48.4226	0.23277	-0.8934
25385.723	276816.48	0.010919	66.0881	27.6035	-172.6100	764.893	27.9192	-55.7132	10	0.058	3102.334	742	53.1334	0.31309	-1.0619
25312.736	276487.05	0.012501	68.0387	29.2729	-168.1258	827.393	29.5195	-52.3124	11	0.065	3102.472	742	57.4141	0.39835	-1.2212
25192.825	275945.33	0.015218	71.0429	31.3669	-161.6306	915.071	32.0909	-47.3698	12	0.076	3102.665	742	63.3965	0.53779	-1.4503
25065.422	275369.03	0.018293	73.9540	32.9571	-155.6785	992.856	34.5153	-42.5209	13	0.087	3102.827	742	68.6776	0.68088	-1.6537
24921.210	274714.31	0.022109	76.9125	34.2142	-149.8765	1066.776	37.0347	-37.4821	14	0.101	3102.986	742	73.6691	0.83320	-1.8409
24755.053	273956.23	0.026925	79.9104	35.1743	-144.1871	1137.895	39.8377	-31.8762	15	0.117	3103.151	742	78.4416	0.99362	-2.0071
24581.150	273157.91	0.032915	82.6288	35.8034	-139.1597	1200.	43.1359	-25.2797	16	0.136	3103.314	728	82.5806	1.14330	-2.1314
24557.221	273048.78	0.033596	82.9702	35.8672	-138.5365	1207.669	43.1097	-25.3321	17	0.137	3103.311	742	83.0896	1.16230	-2.1446
21867.396	250000.	0.241411	100.5414	35.1385	-108.2395	1594.078	55.	-0.	18	0.503	3286.559	705	107.6597	2.05396	-1.7438
21684.159	248784.52	0.230946	101.0915	34.9887	-107.2819	1607.821	55.	-0.	19	0.523	3287.632	748	108.4578	2.07802	-1.7150
20875.867	243461.46	0.246559	103.1750	34.2321	-103.0193	1669.942	49.3795	-15.0655	20	0.533	3287.476	742	112.0472	2.18988	-1.9197
19999.795	237250.58	0.308954	104.8454	33.3866	-98.9497	1732.917	47.5545	-18.7154	21	0.606	3287.313	742	115.5327	2.31919	-2.3774
19066.374	230065.78	0.383434	105.9660	32.5244	-95.2317	1794.089	44.8915	-24.0415	22	0.684	3287.101	742	118.7683	2.47299	-3.1445
18972.144	229305.62	0.391481	106.0416	32.4404	-94.8848	1800.	44.6543	-24.5158	23	0.693	3287.065	728	119.0726	2.48995	-3.2421

17970.853	223773.31	0.491229	106.4045	31.5930	-91.[illegible]39	1860.371	41.4805	-30.8634	24	0.781	3286.811	742	122.0906	2.69241	-4.5568
16846.211	210101.21	0.439841	105.8128	30.7665	-88.1402	1923.114	38.0115	-37.8013	25	0.896	3286.433	742	125.0414	2.97866	-6.7237
15950.587	200642.97	0.795335	104.5374	30.2270	-85.8439	1969.226	36.2909	-41.2426	26	1.051	3286.714	735	127.0762	3.25756	-9.0538
15883.934	200000.	0.785192	104.4189	30.1932	-85.[illegible]	1972.357	36.0678	-41.6888	27	1.059	3286.514	765	127.2098	3.27904	-9.2409
10810.620	150000.	0.434070	85.3957	29.2346	-7[illegible]4	2187.669	22.7061	-40.8776	28	1.457	3070.616	765	134.6229	5.75966	-32.1019
9964.997	148890.91	-0.	81.4829	29.3456	-75.[illegible]88	2222.042	22.7061	-37.3542	29	1.297	2896.583	718	135.4285	6.30612	-36.4704
9707.566	148929.12	0.	80.2937	29.3061	-75.3317	2233.501	22.7061	-36.2815	30	1.230	2837.280	719	135.6743	6.49111	-37.7975
7956.041	135956.62	1.300585	69.6153	30.0570	-72.4268	2345.418	22.7061	-26.9002	31	1.	2470.192	701	137.5847	8.28378	-49.6758
6319.113	122548.58	0.799141	64.0649	30.4405	-71.3864	2397.625	22.7061	-22.1588	32	0.750	2246.075	701	138.1848	9.06022	-55.6608
6268.525	129359.91	0.800905	63.8244	30.4584	-71.3440	2400.	22.7061	-21.9522	33	1.195	2234.786	728	138.2079	9.09386	-55.9181
5174.688	122336.66	1.895406	58.2918	30.4024	-70.4132	2459.147	22.7061	-17.1862	34	5.	1984.258	702	138.6876	9.87641	-61.8467
5000.	121152.80	2.073947	57.6823	30.9497	-70.3213	2465.899	22.7061	-16.6667	35	1.103	1959.418	703	138.7317	9.95865	-62.4958
4940.924	121546.84	2.153491	57.3931	30.9729	-70.2788	2469.096	22.7061	-16.4205	36	0.500	1947.728	701	138.7518	9.99704	-62.8034
4015.524	110635.25	2.886049	52.8822	31.3142	-69.7046	2518.194	22.7061	-12.5647	37	4.	1753.617	702	139.0078	10.53974	-67.5657
4000.	110471.34	2.891767	52.8077	31.3195	-69.6963	2519.005	22.7061	-12.5000	38	1.166	1749.992	703	139.0113	10.54792	-67.6438
3096.098	100926.40	3.769499	48.5957	31.6162	-69.2707	2567.686	22.7061	-8.7375	39	0.250	1514.011	701	139.1760	10.98794	-72.0462
3019.061	100000.	3.962067	48.2442	31.6410	-69.2379	2572.182	22.7061	-8.4128	40	1.091	1491.979	765	139.1876	11.02354	-72.4137
3000.	99767.73	4.013168	48.1582	31.6471	-69.2299	2573.292	22.7061	-8.3333	41	1.089	1486.573	703	139.1904	11.03219	-72.5033
2971.535	99416.39	4.092393	48.0300	31.6561	-69.2181	2574.957	22.7061	-8.2147	42	3.	1478.490	762	139.1945	11.04509	-72.6370

ENDURE = 0.000 912.000 = TNCNTS

THE CALCULATION FOR CASE NUMBER 3312 HAS BEEN TERMINATED. TIME USED = 1 MINUTES, 36.267 SECONDS.

ADDITIONAL OUTPUT FOR CASE NUMBER 3312

NO.	CNT	COUNT	NSTP	NUS	FLIGHT DISTANCE	BETA	MACH NO	ACCEL	TEMP NO EQ	TEMP1	TEMP2	TEMP3	TEMP4	TEMP5	TEMP6
1	792	0	0	0	0.	0.0000	29.412	0.0024	1915.412	1000.00	1000.00	1000.00	1000.00	1000.00	1.00
2	705	163	8	8	3246050.50000	8.7513	29.438	0.0354	2682.115	2248.55	2544.45	2351.83	2236.35	1515.61	2.71
3	748	267	1	17	6555618.75000	17.6914	29.152	0.1430	3177.035	3155.72	3397.59	3044.51	3036.82	2659.63	7.70
4	718	402	15	14	6806697.99023	18.3699	29.136	0.0368	3176.616	3165.13	3405.91	3049.46	3044.75	2721.99	8.18
5	718	461	19	11	13663208.34521	36.8986	28.947	0.0313	3102.301	3104.23	3335.20	2982.31	2983.54	2990.51	20.51
6	749	464	2	12	13958152.34521	37.6956	28.940	0.0313	3101.995	3103.12	3334.26	2981.53	2982.49	2988.60	21.02
7	728	619	11	8	15472495.59521	41.7876	28.891	0.0444	3102.011	3102.01	3333.34	2981.36	2981.41	2983.59	23.63
8	742	635	4	6	16042735.59521	43.3285	28.870	0.0441	3101.991	3102.01	3333.34	2981.37	2981.41	2982.80	24.61
9	742	681	6	4	17924303.59521	48.4238	28.796	0.0508	3102.167	3102.16	3333.50	2981.53	2981.56	2981.82	27.87
10	742	726	7	3	19671791.59521	53.1352	28.717	0.0576	3102.334	3102.32	3333.67	2981.69	2981.72	2981.73	30.89
11	742	771	6	1	21256175.59521	57.4167	28.635	0.0646	3102.472	3102.46	3333.82	2981.83	2981.85	2981.80	33.65
12	742	825	3	4	23470447.59521	63.4004	28.499	0.0756	3102.665	3102.65	3334.02	2982.01	2982.03	2981.96	37.51
13	742	878	6	2	25425263.59521	68.6832	28.355	0.0874	3102.827	3102.81	3334.19	2982.16	2982.19	2982.12	40.94
14	742	937	4	3	27272941.59521	73.6766	28.192	0.1007	3102.986	3102.96	3334.36	2982.31	2982.34	2982.27	44.20
15	742	985	5	2	29039597.59521	78.4512	28.004	0.1166	3103.151	3103.13	3334.54	2982.47	2982.49	2982.42	47.33
16	728	1028	8	11	30571773.84521	82.5922	27.807	0.1362	3103.314	3103.27	3334.69	2982.61	2982.63	2982.55	50.07
17	742	1036	2	5	30700189.84521	83.1014	27.780	0.1366	3103.311	3103.28	3334.70	2982.62	2982.64	2982.57	50.41
18	705	1156	8	12	39849994.09521	107.6776	23.781	0.5035	3286.559	3281.66	3527.50	3156.62	3154.26	3133.17	68.76
19	748	1161	3	15	40144915.09521	108.4756	23.495	0.5234	3287.632	3285.24	3530.70	3159.04	3157.63	3140.48	69.51
20	742	1317	5	4	41470995.09521	112.0642	22.264	0.5326	3287.476	3287.48	3532.55	3159.65	3159.68	3155.35	72.97
21	742	1408	3	4	42758547.09521	115.5495	20.952	0.6065	3287.313	3287.35	3532.46	3159.50	3159.55	3158.65	76.47
22	742	1497	5	4	43953811.09521	118.7860	19.580	0.6837	3287.101	3287.15	3532.25	3159.30	3159.35	3159.22	79.87
23	728	1509	7	11	44066231.59521	119.0904	19.443	0.6928	3287.065	3287.13	3532.23	3159.28	3159.33	3159.23	80.20
24	742	1600	5	3	45181751.59521	122.1121	18.005	0.7805	3286.811	3286.87	3531.96	3159.03	3159.09	3159.16	83.55
25	742	1682	4	3	46274487.59521	125.0734	16.430	0.8958	3286.433	3286.52	3531.57	3158.67	3158.75	3158.89	87.03
26	735	1742	2	4	47030967.59521	127.1244	15.233	1.0513	3286.714	3286.36	3531.44	3158.60	3158.61	3158.68	89.59
27	765	1751	8	11	47080804.22021	127.2595	15.157	1.0590	3286.514	3286.41	3531.51	3158.66	3158.66	3158.69	89.76
28	765	1839	11	7	49972386.47021	135.1098	10.075	1.4572	3070.616	3127.10	3349.86	2990.70	3002.72	3057.98	101.08
29	718	1866	17	11	50329184.68408	136.0798	9.373	1.2969	2896.583	2988.11	3193.24	2846.95	2866.85	2965.89	102.38
30	718	1881	10	16	50441891.19531	136.3863	9.063	1.2301	2837.280	2935.82	3135.63	2795.10	2815.56	2928.33	102.75
31	701	1933	15	7	51402808.17188	138.9991	7.109	1.2628	2470.192	2556.21	2729.97	2432.51	2451.07	2593.02	105.33
32	701	1967	13	6	51762251.10938	139.9767	6.088	1.2027	2246.075	2393.77	2548.49	2267.75	2291.79	2457.72	106.11
33	728	1976	9	7	51777200.60938	140.0174	6.043	1.1950	2234.786	2385.37	2539.07	2259.23	2283.48	2451.21	106.14
34	702	2004	8	13	52113428.78125	140.9322	5.000	1.0977	1984.258	2175.77	2304.99	2045.50	2078.64	2289.75	106.69
35	703	2015	9	11	52147611.50000	141.0252	4.888	1.1034	1959.418	2153.58	2280.53	2023.66	2057.05	2272.07	106.74
36	701	2027	12	11	52163500.73828	141.0685	4.836	1.1070	1947.728	2143.29	2269.27	2013.60	2046.98	2263.76	106.76
37	702	2052	12	9	52383518.16016	141.6669	4.000	1.1659	1753.617	1991.98	2102.43	1862.32	1899.88	2142.04	107.04
38	703	2060	8	15	52446776.28906	141.6758	3.986	1.1660	1749.992	1989.55	2099.75	1859.92	1897.49	2140.09	107.04
39	701	2082	8	13	52 9174.43750	142.1447	3.124	1.0997	1514.011	1842.24	1934.82	1709.27	1753.17	2025.93	107.21
40	765	2093	10	8	52 2923.62500	142.1821	3.047	1.0913	1491.979	1826.83	1919.75	1695.55	1739.94	2015.67	107.22
41	703	2102	9	14	52 6263.76367	142.1912	3.028	1.0893	1486.573	1825.54	1916.07	1692.21	1736.68	2013.14	107.22
42	762	2113	11	10	52581235.33398	142.2047	3.000	1.0865	1478.490	1820.60	1910.53	1687.18	1731.81	2009.36	107.23
	TOTALS		962	824			ENDURE =	0.0000	917.000 = TNCNTS						

LINE NUMBER	28	19	20	20	20	20	23	42
MAXIMUM VALUE	1.4572	3287.632	3287.48	3532.55	3159.65	3159.68	3159.23	107.23

VALUES USED	RTRADS =	2.00000	1.50000	1.50000	1.87500	1.50000	1.00000
TO COMPUTE	CMOS =	0.50000	0.50000	0.50000	0.50000	2.00000********
TEMPERATURES	E =	0.50000	0.50000	0.78125	0.62500	0.78125	0.00000

EQUILIBRIUM RATIOS ARE		1.00000	1.07457	0.96112	0.96112	0.96112	0.00000
MAXIMUM EQUILIBRIUM TEMPERATURES ARE		3287.63	3532.79	3159.82	3159.82	3159.82	0.00

END OF OUTPUT FOR CASE NUMBER 3312

Unclassified
Security Classification

DOCUMENT CONTROL DATA · R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, Canada	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

AN ATTITUDE CONTROL SYSTEM TO CONSTRAIN THE SKIN TEMPERATURE OF A MANNED LIFTING SPACECRAFT DURING REENTRY INTO THE EARTH'S ATMOSPHERE

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Scientific

5. AUTHOR(S) (Last name, first name, initial)
Fine, Jerome H.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 1967	145	40

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
AFOSR 222-66 b. PROJECT NO.	UTIAS REPORT NO. 126
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFOSR 67-0858
d.	

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Office of Scientific Research 1400 Wilson Blvd., Arlington, Virginia 22209

13. ABSTRACT

An attitude control system to regulate the temperature of a manned lifting spacecraft during reentry into the Earth's atmosphere is proposed. Its use prevents the peak skin temperature that is experienced during the reentry from rising moderately beyond that which would occur during an equilibrium glide of the same vehicle. The effects of Earth rotation and oblateness upon the performance of the attitude control system were found to be moderate and predictable. The maximum temperature increment associated with them was found to be only 100F° for the worst set of initial conditions. The cross range shift of the footprint due to rotation was found to be within 70 miles of the value that would occur for the corresponding orbit in vacuum. Oblateness could generally be accounted for by using the effective initial glide angle relative to the Earth's surface rather than the geocentric initial value relative to the central coordinate system. The results of density variation in the Earth's atmosphere were not serious. Large increases in the maximum skin temperature occurred only when extremely large spatially random density disturbances were encountered by the vehicle.

DD FORM 1 JAN 64 1473

Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Reentry - heating Reentry - lifting body Reentry - temperature control Reentry - effect of earth oblateness Reentry - effect of density perturbations						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional

GPO 886-551